

# 电脑编程技巧与维护

COMPUTER PROGRAMMING SKILLS & MAINTENANCE

<http://www.comprg.com.cn>

上  
3月

2013年3月03日

每期定价:11.00元 全年定价:264.00元  
《电脑编程技巧与维护》杂志社出版

刊号: ISSN 1006-4052  
CN 11-3411/TP

广告许可证 京海工商广字0151

国家级科技期刊 中国学术期刊综合评价数据库统计源期刊 中国核心期刊(遴选)数据库收录期刊

 [www.directui.com](http://www.directui.com)  
**DirectUI 界面库**

免费咨询热线: 400-660-9989

—— 让界面与业务逻辑彻底分离

易学易用、缩短80%的界面开发周期、提升界面运行效果与质量

成功应用在华为、中兴、盛大网络、中国移动、铁道研究院、瑞星、步步高等知名企业

第95页:《UIPower成功助力新奥特完成喜马拉雅非编软件的界面革新项目》



 **UIPower**  
[www.uipower.com](http://www.uipower.com)

ISSN 1006-4052



LAICAR.COM

shop35833438.taobao.com





51CTO 传媒

# 2013大数据全球技术峰会

## Big Data Global Summit 2013

2013/4/26-2013/4/27 北京富力万丽酒店 三层

官网: <http://wot.51cto.com/bigdata2013>



抢票热线

010-68478816

企业团购

010-68479366

立即购票 直降400元  
企业团购 聚优惠

# 让数据发出声音!

# Sound

2013大数据全球技术峰会将围绕大数据  
**基础架构与上层应用**的生态系统,  
解决大规模数据引发的问题,探索大数  
据基础的解决方案,激发数据挖掘带来的  
竞争力。



扫一扫快速购票

30余位海内外资深技术专家  
**六大主题论坛**  
Hadoop生态系统及分布式架构设计  
NoSQL and NewSQL  
企业应用与大数据

云计算与大数据  
数据整合与挖掘分析



2013年第05期  
3月(上)

# 电脑编程技巧与维护

总第275期 1994年7月创刊 (半月刊)

社长: 孙茹萍

副社长: 田真

总编: 王路敬

编辑委员会

主任: 梁祥丰

委员: 胡顺增 刘江 莫亚柏

(拼音为序) 孙春亮 温莉芳 吴淑珍

严晓舟 张立荣

编辑: 侯穆蕾 姬振伟

刘艳彬 杨月慧

美编: 范志飞

公关部主任: 苏加友

出版发行部: 刘文海

编辑出版: 《电脑编程技巧与维护》杂志社

主管部门: 中华人民共和国工业和信息化部

主办单位: 中国信息产业商会

地址: 北京市海淀区长春桥路5号  
6号楼1209室

投稿邮箱: gaojian@comprg.com.cn

gaojian@comprg.sina.net

编辑部信箱: gaojian@comprg.com.cn

发行部信箱: zzsfx@vip.sina.com

网址: http://www.comprg.com.cn

邮编: 100089

电话: (010) 82561037

传真: (010) 82561614

照排: 《电脑编程技巧与维护》  
杂志社电脑排版部

印刷厂: 北京慧美印刷有限公司

订阅处: 全国各地发行局

国内总发行: 北京报刊发行局

邮发代号: 82-715

国外发行代号: M6232

ISSN 1006-4052

刊号: CN11-3411/TP

广告订可证: 京海工商广字 0151号

全年定价: 264元

每期定价: 11元

## 飞天Rockey加密锁

飞天诚信  
我们构筑安全

### 引领“智能·低价”风暴

- 震撼价格, 超高性价比

- 智能卡芯片

- 无驱, 使用更方便

- 涵盖高、中、低端产品



系统支持:

Windows 98SE/Me/2000/XP/Server 2003/2008/Vista/7/8, Linux, \*MacOS等多平台

飞天诚信科技股份有限公司

地址: 北京市海淀区学清路9号汇新大厦B座17层 邮编: 100085  
华南营销中心: 020-38870851 华东营销中心: 021-58202298

www.FTsafe.com

电话: 010-62304496

传真: 010-62304477

西南营销中心: 028-85481711 华中营销中心: 027-87160151



## 域天32位智能卡



36元

专为共享软件作者设计, 使得共享软件作者实现零成本加密!

- 硬件32位智能卡(内置32位CPU)及专有防克隆技术;保证无法复制
- 软件代码在智能卡中运行, 内置硬件3DES及RSA算法, 无法破解
- 全速USB协议, 传输速度高达12Mbps
- 先进的动态加密技术, 加密代码不受长度限制
- 支持多种开发语言, 在加密锁中可以运行跳转, 比较, 循环, 查表, 函数调用等指令及字符串操作
- 超大容量内部存储器: 30K字节独立储存空间
- 易于使用的编译及调试器, 专有的代码生成器及模糊解释语言, 方便开发商进行开发
- 内置时间模块, 支持时间限制功能
- 授权锁模式, 使得软件的代理销售更容易控制

东莞市域之天软件开发有限公司

电话: 0769-22686137 传真: 0769-22688320

Http://www.dgyzt.com

E-mail: ytkj\_911@163.com



来卡网出品

LAICAR.COM

shop35833438.taobao.com



稿件一经采用，即寄样刊，版权归杂志社所有。本刊图、文版权所有，未经允许不得任意转载和摘编。

### 新技术追踪

- 4 WebKit 引擎一统江山，网络将成为另一个 App Store 等两篇

### 跟高手学编程

- 5 iBatis.Net (C#) 系列一：简介及运行环境 ..... 张德强 祁亚玲  
简介 iBatis.Net 的运行环境、运行原理，并通过实例，详细讲解在 VS 2012 下建立的 C# 项目中 iBatis.Net 的使用方法。

### 编程语言

- 9 基于 LabVIEW 的声卡双通道数字存储示波器  
..... 马海瑞 韩云东 郑振宇  
基于 LabVIEW 的模块化编程，设计并实现了具有对音频信号的测量、存储和回放等功能的声卡双通道数字存储示波器。
- 12 VC2010 制作单词、双词同现的频率统计工具 ..... 马创新  
介绍使用 VC++ 2010，制作具有“词语频率”、“词性频率”、“双词频率”和“双词性频率”等词语和词性频率统计功能工具的方法。
- 17 Java 开发目录同步程序  
..... 江洪  
通过讲解 Java 语言套接字，实现 C/S 结构的目录同步程序开发的思路与方法。
- 24 对酒店音响设备自动开关的实现  
..... 刘同法  
利用 C++ 编程，实现对酒店吧台与包厢音响设备的自动开关，从而达到节省用电的目的。

### 专家论坛

- 27 Android 平台 GPS 系统的应用开发 ..... 庞国明 隋瑞红

讲述利用 Android 平台的通用框架，开发基于智能手机的 GPS 导航应用软件的方法。

### 数据库

- 31 Oracle 程序包按自然方式自动维护分区 ..... 刘恒学  
通过编写 Oracle 程序包，实现按照自然方式自动维护分区的方法，以达到减少人工维护量和空闲空间占用量的目的。
- 38 用 Excel VBA 灵活复制照片并修改文件名 ..... 张贻忠 洪冰  
日常办公中，用 Excel VBA 编写程序，复制照片，并同时修改文件名，降低因人工录入、修改时可能出现的错误。

### 网络与通信

- 47 ASP.NET 服务器日历控件的开发和使用 ..... 郭海伟  
通过一个基于 ASP.NET 的服务器日历控件的开发，阐述了开发、调试和使用 ASP.NET 服务器控件的方法。
- 52 JavaScript、ASP 车辆调度系统  
..... 商涛  
利用 ASP 和 JavaScript，开发一个基于局域网 B/S 架构的企业车辆调试系统。
- 55 邮件查看软件设计  
..... 刘步林  
利用 Delphi 的控件编程，设计并实现邮件的查看。

### 图形图像处理

- 62 DirectShowLib 的家庭视频监控系统 ..... 陶阳  
基于 DirectShowLib 类库，设计并实现了一个家庭视频监控系统原型，通过对 USB 无驱动摄像头的操控，实现视频的采集、抓图和录像功能。





# 目次

## 实用第一

## 智慧密集

敬告读者：邮政部门独家代理发行本刊，未委托小蓝帽发行公司及其他社会公司办理本刊订阅业务。特此声明！

PERFECTION IN SOFTWARE PROTECTION

CodeMeter

CodeMeter 软件加密解决方案 - 安全、易用、灵活！

免费试用

- 自动加密C++，Delphi，VB等程序，.net程序集，Java程序集
- 兼容Win2000, WinXP, Vista, Win7, WinCE, WinARM, Linux, LinuxARM, MacOS, VxWorks等操作系统
- 全自动加密，无需任何代码编程
- 按需加解密原理，高安全，多次黑客大赛中无任何破解
- 32位智能卡，内置64k/384K存储空间，无需安装驱动
- 内置时钟功能，设定激活时间、过期时间、使用天数、维护时间
- 授权狗管理模式，加密狗与授权分开管理，提高管理安全性
- 灵活的授权管理：分模块管理、版本管理、单机网络管理
- 无需任何代码开发，实现方便的远程升级功能
- 德国研发、制造，符合CE、FCC、VCCI、DVE、UL、BAFA、RoHs等国际认证



申请办法：请登录我们的申请网页：<http://www.wibu.com.cn/sdk.php.htm>

 咨询热线：021-55661790（上海） 010-82961560（北京）

WIBU  
SYSTEMS

威步信息系统(上海)有限公司 <http://www.wibu.com.cn> Email:Sales@wibu.com.cn

- 67 用 VB 实现 PNG 四态按钮控件  
..... 曹卫忠  
利用 VB 编程，实现 PNG 格式四态按钮控件的方法。

### 游戏编程

- 70 C# 语言开发贪吃蛇游戏  
..... 王文举  
利用 C#，创建独立功能类，实现一个结构清晰，扩展方便的贪吃蛇游戏。

### 计算机安全与维护

- 73 手机基站定位的实现与应用  
..... 汪永松  
在 Android 平台，基于手机基站，实现了低成本的定位服务，并通过开发实例，对定位的应用进行详细说明。
- 82 Visual Assist X 与 VBA 结合实现在 Word 里排版代码  
..... 刘 烽  
利用 Visual Assist X VS 插件与 VBA

编程相结合的方法，解决了 Word 里排版代码质量不高的问题。

- 84 Delphi 开发中的复用技巧  
..... 张扬嵩  
解析一种在 Delphi 数据库应用开发中有效复用的方法和技巧。

### 编程疑难问题解答

- 90 如何用 OpenLu 解方程  
..... 王禄 冯志广 吴阳 潘辉
- 92 怎样处理 C# 开发中常见问题  
..... 张伟强

### 博士信箱

- 94 电脑系统维护经验与技巧  
为您服务
- 96 新书点评



来卡网出品  
LAICAR.COM  
[shop35833438.taobao.com](http://shop35833438.taobao.com)



## WebKit 引擎一统江山，网络将成另一个 App Store

作为独立浏览器中硕果仅存的代表之一，Opera 也踏上了变革之路。很快，Opera 就将弃用其自主渲染的 Presto 引擎，改用 WebKit 引擎。

时下人气最旺的 GoogleChrome 浏览器以及苹果的 Safari 浏览器采用的也是 WebKit 引擎。而在移动端最流行的浏览器——iPhone 和 iPad 上的移动版 Safari，以及安卓系统的默认浏览器，同样基于 WebKit 技术。换句话说，WebKit 已经悄无声息的完成了浏览器领域的“大一统”。

可以毫不夸张的说，WebKit 在这场战役中已经大获全胜。

这意味着不久之后创建网站将变得和过去大不相同——无需费力设计出能兼容四款不同引擎、几款不同浏览器的网站了，只需创建出能在 WebKit 上工作的网站即可。过去是利用网络技术创建网站，现在则是利用 WebKit 技术开发应用。换句话说，这一流程将变得越来越像是为应用商店开发应用，比如为 iPhone 开发应用且无需人工审核。

这并非意味着“网络即应用”的时代已经到来，但我们已经知道，它终将变为现实。

对于花了几十年时间纠结于什么属于 HTML、什么不属于 HTML 的机构来说这无疑是一个极大的危机。WebKit 渐渐占据统治地位，意味着其最大的客户 Google 和苹果将会定义什么属于网络、什么不属于网络。在这次变革中走在队伍最末尾的是微软。IE6 一度是全球最流行的浏览器，微软也致力于将专属网络标准推广成主流。

或许对于 WebKit 成为主流这件事，最有力的解读莫过于微软的担忧了。11 月份微软写给开发者的博客中说：“你们或许只在针对 Android 或 iOS 进行优化的网站上使用 WebKit。”

不过现在，微软的声明中透着些许绝望，“让 WebKit 优化的网站支持 IE10 也不是什么难事。”

这则声明没有说明微软何时会进行转变。但可以肯定的是，微软已经别无选择。

### 互联网走向“世界数据流”

人们总是好奇地发问：“下一代互联网会是什么样子？然而事实却是，不会有下一代互联网了。”这是耶鲁大学计算机科学教授大卫·格勒恩斯特在一次采访中说的。

我们现在所使用的以空间为基础的网络将会逐步被以时间为基础的“世界数据流”（worldstream）所代替。这一幕已经发生，并以“人生数据流”（lifestream）的形式为展开。

“人生数据流”是一种用不同种类信息组成的、内容可搜索的实时信息流，目前的博客、RSS feeds、Twitter、Facebook 的 Walls 和 Timelines 以及其他聊天数据流都属于这一类别。其架构突显了从“台式机平原模式”向更加纵深的可以代表具体

时间点的数据流方向发展。

这就好比从台式机过渡到一本魔幻的日记：我们可以记录下每一页日记的内容，而日记又会自动翻页，这样就可以追踪到自己每时每刻的行动。直到拿起日记，翻页就会停止。从某种程度上说，这本日记就变成了人生的参考书：一本完整的可追溯的人生指南。如果把它放下，日记就会开始继续翻页。

现在，这种如日记般的架构正在取代空间成为互联网领域的主要标准。网络上的所有信息都会很快成为基于时间架构的信息。在以字节为标准的世界里，基于空间的架构处于静止状态。而基于时间的架构则更具活力，更具流动性，这与时间的特点完全一致。

### 互联网将成为历史

截至目前，我们所使用的互联网一直都是以空间为基础架构起来的，我们在计算领域积累了 20 多年的经验才建设成如今的网络。

当前，互联网最重要的功能就是发布最新消息，让人们知道时下正在发生的事情。当然，我们仍然可以对过去进行浏览或者搜索，因为时间总是在网络空间来回穿梭。任何信息客体都能够在“当前”添加，然后慢慢地成为过去的的数据。

如果我们把所有这些博客、信息源、聊天数据等都整合在一起，那又会是什么样子呢？如果把网络上的每一个时间流都汇总成一股数据流的话，我们就得到了“世界数据流”：一种将网络空间视为一个整体的方式。

没有人能够真正窥探到整个“世界数据流”的全貌，因为多数流过网络的信息都是属于个人的。不过，每个人都可以看到其中的一部分。

目前的网站都是静态的，而信息则将通过“世界数据流”不断且稳定地流向过去的时间点。那这一切又意味着什么呢？

### 数据流彻底改变搜索市场

现在的操作系统、浏览器以及搜索模式都将变得过时，因为人们再也不想被电脑或网站所束缚。

人们真正想要的是接入信息。既然网络空间里存在着无数个短暂的单独“人生数据流”，那么基本软件就将转为“数据浏览器”，它将帮助找到并接入我们想要的“人生数据流”。

如今的网络架构方式让我们足不出户就可以获得访问数千家实体商店的体验。以时间为基础的“世界数据流”使我们坐在家就能够纵览网络上个性化的流行趋势。

这种精确的控制力限制了网络天生具有的偶然性，这一未来发展趋势不仅扼杀了我们所熟知的操作系统、浏览器和搜索，而且还改变了我们熟知的“计算”的含义。无论设备大小，计算机在不远的将来所发挥的主要功能就是接入不断流动的的全球网络数据流，这就好比车载收音设备接收广播电台信号一样。我们将更加关注信息世界以及我们与之相联系的生活，因此对计算设备本身将变得不再那么关注。





# iBatis.Net (C#) 系列一：简介及运行环境

张德强 祁亚玲

**摘要：**介绍 iBatis.Net 的原理和运行环境中各参数的配置情况，并通过实例，详细讲解通过 VS2012 建立的 C# 项目中 iBatis.Net 的使用。

**关键词：**iBatis.Net 工具；C# 语言；运行环境；实例

## 1 iBatis.Net 简介

iBatis 一词来源于 “Internet” 和 “abates” 的组合，是一个由 Clinton Begin 在 2001 年发起的开放源代码项目，现已成为以 SQL 为中心的持久化层框架。iBatis.Net 是从 iBatis 的 Java 版本移植过来的 .NET 版本。目前项目地址为：<http://code.google.com/p/mybatisnet/>，可以通过项目网站下载其源代码。

iBatis 提供的持久层框架包括 SQL Maps 和 Data Access Objects (DAO)，其中最主要的是 SQL Maps。和其他对象关系映射 ORM，如 Hibernate 和 Spring 等“一站式”对象关系映射解决方案而言，iBatis 则是一种“半自动化”的对象关系映射实现，它允许程序员自己控制 SQL 语句，为系统设计提供了更大的自由空间。

### 1.1 SQL Maps

SQL Maps 是这个框架中最重要的部分，它是整个 iBatis Database Layer 的核心价值所在。通过使用 SQL Maps 你可以显著地节约数据库操作的代码量。SQL Maps 使用一个简单的 XML 文件来实现从实体到 SQL 语句的映射。跟其他的框架或者对象关系映射工具相比，SQL Maps 最大的优势就是简单。同时它需要学习的东西很少，在连接表或复杂查询时也不需要复杂的模式，使用 SQL Maps，可以自由地使用 SQL 语句。这种独特的数据映射模式（即 Data Mapper），使用 SQL 映射的方式将对象持久化至关系型数据库。简单的理解就是它将在数据访问层实现的 C# 逻辑代码，变为通过关系数据库与对象的映射来实现，并将 SQL 逻辑放到外部的 XML 配置文件中，以方便以后的维护。

### 1.2 Data Access Objects (DAO)

Data Access Objects 允许通过一个简单接口的来操作数据，从而隐藏了实现的细节。使用 DAO，可以动态配置应用程序来访问不同的实体存储机制。实际上 DAO 通过一个通用的 API 层把具体操作数据的细节封装了起来，不必再关心操作数据的细节，而把中心放在业务逻辑上，避免了 ADO.NET 的复杂性。如果有一个复杂的应用需要用到几个不同的数据

库，DAO 可以建立一个一致的 API，让系统的其他部分来调用。实现了应用系统的数据库无关性，是不是有点工厂模式的味道。

## 2 SQL Maps 原理

提到 SQL Maps 原理，不得不提到官方帮助文档给出的一副图，如图 1 所示。

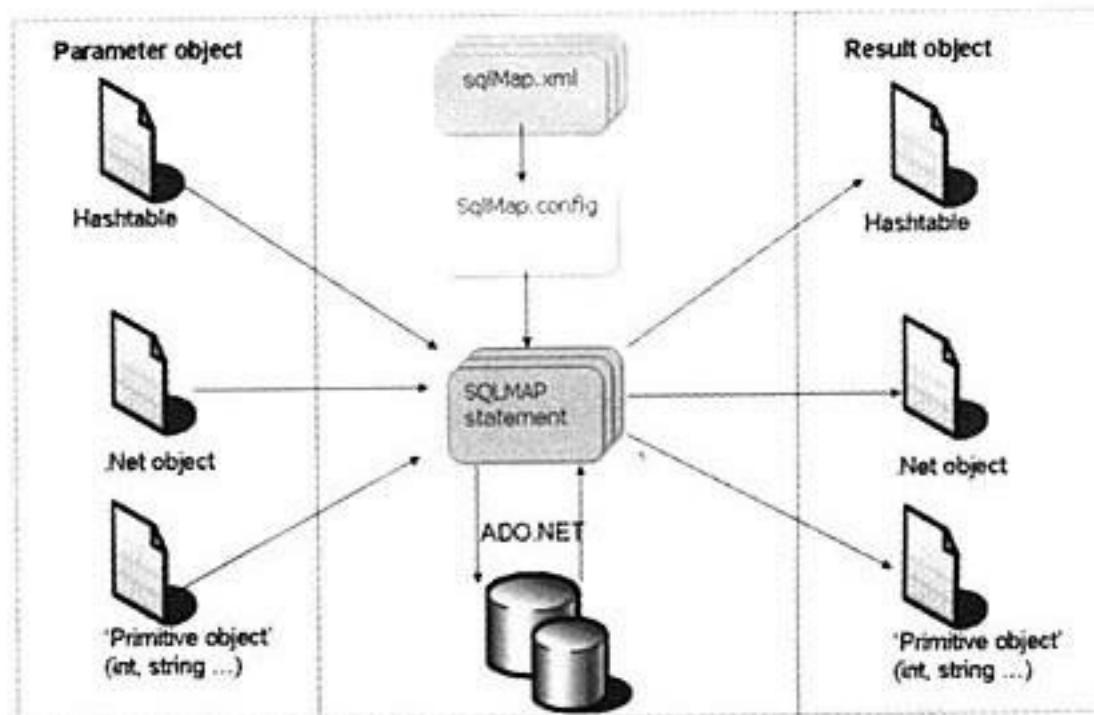


图 1 iBatis 的数据映射工作流程图

该图清晰地反映了 iBatis 的数据映射工作流程。左边是输入参数，中间为记录在 SqlMap.xml 文件里面的 SQL 映射语句被 SQL Maps 组件通过 SqlMap.config 配置文件读入，通过组件内封装的 ADO.NET 和数据库进行交互。输出右边的结果对象。

整个开发的重点就在 XML 描述的编写上，其他的工作由 SQL Maps 组件来完成，基本的步骤如下：

(1) 程序输入一个参数，无论是对象还是一个原生类型。参数将被用于设置 SQL 语句或存储过程的运行时的值

(2) 通过传送参数和在 XML 描述中的声明名字或者存储过程来执行映射。这一步将是关键的步骤。iBatis 框架将会准备 SQL 语句声明或者存储过程，用参数设置运行时数据值，即带入运行数据值，执行 SQL 语句或者存储过程，返回结果。





(3) 和 ADO.NET 一样，当数据更新的时候，更新的行数将会被返回。当数据查询的时候，返回的将是一个对象或者对象的集合。结果对象或者对象的集合，可以是一个普通对象或者原生类型，通过 XML 描述中的参数来指定。

## 3 运行环境

iBatis 基本的运行环境配置主要由两个文件组成，分别是 SqlMap.config 和 providers.config。它们是必需的两个配置文件，其中 SqlMap.config 的功能类似于应用程序的 web.config 或者 app.config，是 iBatis 核心的配置文件，靠它把 XML 描述文件载入给 iBatis 框架。默认情况下，需要把该文件放在应用程序的运行目录下并且保留它的文件名称。如果改变名称或者存放在其他目录，需要在程序里面额外手工指定。

providers.config 是一个数据驱动提供类的配置文件，提供了如常见的 Oracle、Sql Server、OleDb、Odbc、MySQL 和 Informix 等数据库的数据驱动，其他数据库可以通过增加相应的配置来添加。

SqlMap.config 的结构如图 2 所示，包括以下一些主要的配置节，根据需要，有的配置节并不是必须的。

```

Toolbox
default.aspx  SqlMap.config  X default.aspx.cs
<?xml version="1.0" encoding="utf-8"?>
<sqlMapConfig
  xmlns="http://ibatis.apache.org/dataMapper"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <settings>
    <setting useStatementNamespaces="false" />
    <setting cacheModelsEnabled="true" />
    <setting validateSqlMaps="true" />
  </settings>
  <!-- 数据库连接配置信息 -->
  <providers resource="providers.config"/>
  <database>
    <provider name="oracleClient1.0"/>
    <dataSource name="db" connectionString="Data Source=(DESCRIPTION=
      (ADDRESS=(PROTOCOL=TCP)(HOST=172.29.21.37)(PORT=1521))(CONNECT_DATA=
      (SERVICE_NAME=orcl.moban.com)));User Id=xxzx;Password=dean;"/>
  </database>
  <!-- XML 映射文件配置信息 -->
  <sqlMaps>
    <sqlMap resource="Post.xml" />
  </sqlMaps>
</sqlMapConfig>

```

图 2 SqlMap.config 内容

(1) settings：包括 3 个配置段：

```

<setting useStatementNamespaces="false"/>
<setting cacheModelsEnabled="true"/>
<setting validateSqlMap="false"/>

```

useStatementNamespaces：是否使用语句命名空间，这里的命名空间指的是映射文件中 sqlMap 节点的 namespace 属性。即如果该参数设为“true”，语句调用时需追加命名空间，如：mapper.Update (“TestApp.UpdatePost”, post); 否则直接通过语句名称调用即可，如 mapper.Update (“UpdatePost”, post)。但这时一定要注意，此时需要保证所有映射文件中语句定义没有重

名。可以通过变量来指定该参数设置，如：<setting useStatementNamespaces=" \$ {useStatementNamespaces}" />，参数通过 properties 属性来指定，该值默认为“false”。

cacheModelsEnabled：配置是否启用 iBatis 的缓存机制，默认是“true”。

validateSqlMap：配置是否启用 SqlMapConfig.xsd schema 验证映射文件，默认是“false”。

(2) providers：配置数据驱动提供类配置文件的路径和文件名。可以通过 resource（或 url、embedded，它们分别是引用 url 和编译在程序中的资源文件）属性引用进来。如：<providers resource=" providers.config" />，这 3 种载入方法说明如表 1 所示。

表 1 providers 的参数

参数	描述
resource	指定数据驱动配置文件从项目的根目录进行加载，如 resource="providers.config"
url	指定数据驱动配置文件从文件的绝对路径进行加载，如 url=" c:\IBatis\Resources\providers.config" -或者- url=" file://c:\IBatis\Resources\providers.config"
embedded	指定数据驱动配置文件可以作为程序集的资源文件进行加载，如 embedded=" Resources.providers.config, MyApp.Data"

凡涉及到引用外部文件的都可以通过以上 resource，url，embedded 3 种方式实现。

(3) database：数据库的信息，包括使用哪些数据库驱动和数据库连接字符串的配置。provider 参数定义数据库访问所使用的 provider.config 文件定义的 provider；dataSource 参数定义特定的数据库连接字符串。

(4) SqlMaps：指定 XML 映射文件的位置，配置中可以出现多个 SqlMap 节点，以指定项目内所包含的所有映射文件。如通过 embedded 方式指定 <sqlMap embedded=" sqlFile.xml, ConsoleApplication1" />。

(5) properties：配置一些常量属性，如：<property key="useStatementNamespaces" value=" false" />。如果这些属性有很多的话可以单独写一个文件里面，再通过 resource（或 url，embedded 分别是引用 url 和编译在程序中的资源文件）属性引用进来

(6) alias：类型别名的配置，为了使用更方便的使用类（类名更短），就需在这里进行别名的配置。如<alias><typeAlias alias=" YesNoBool" type=" IBatisNet.Test.Domain.YesNoBoolType HandlerCallback, IBatisNet.Test" /></alias>

(7) typeHandlers：定义数据库类型到 dotnet 数据类型的





## FOLLOW MASTER PROGRAM

处理,不同的数据库都有一些特殊的数据库字段类型需要特殊处理,就可以通过这个功能实现。通过它为该类型取一个名字,并且指定对应的.NET类型来处理它。如 <typeHandlers>  
<typeHandler type=" bool" dbType=" Varchar" callback=" YesNoBool" /> </typeHandlers>

以上就是 SqlMap.config 的基本内容了。Provider.config 文件的配置信息相对就很简单,IBatis.Net 使用的是插件式结构来使用这些数据库提供程序,每一个 Provider 对应于 providers.config 文件中定义的一个 provider 项。提供程序要求安装相关类库,每一个 provider 元素都有" enabled" 属性来控制是否启用这个 providers。

## 4 日志处理

IBatis 框架具有和 Apache Log4Net 一样的内部日志机制,记录了与数据库的交互情况。可以使用框架内置的 3 个记录器 (NoOpLogger, ConsoleOutLogger, TraceLogger), 或者使用像 Apache Log4Net 一样的外置记录包。为了让 IBatis 记录器工作,必须在配置文件里 (App.Config 或者 Web.Config) 配置一些节点:

注册 IBatis 日志处理节点和 log4net 节点:

```
<configSections>
<sectionGroup name="iBATIS">
  <section name="logging" type="IBatisNet.Common.
Logging.ConfigurationSectionHandler, IBatisNet.Common" />
</sectionGroup>
  <section name="log4net" type="log4net.Config.
Log4NetConfigurationSectionHandler, log4net" />
</sectionGroup>
</configSections>
```

配置 IBatis 节点:

```
<iBATIS>
  <logging>
    <logFactoryAdapter type="IBatisNet.Common.
Logging.Impl.Log4NetLoggerFA, IBatisNet.Common.Logging.
Log4Net">
      <arg key="configType" value="inline"/>
    </logFactoryAdapter>
  </logging>
</iBATIS>
```

注意 "<arg key=" configType" value=" inline" />" ,其 Value 值可以有以下几个选项:

inline: log4net 节点在 App.Config 或者 Web.Config 文件中配置。

file: 使用外置配置文件,需要 configFile 参数配合使用, <arg key=" configFile" value=" 外部配置文件"。

file-watch: 与" file" 一样,只是多了一个监视外部配置文件的变动功能,如果有变动则重新加载配置。

external: IBatis 将不会尝试配置 Log4Net。

需要特别注意一下,如果采用 file-watch 外部文件配置的方式时,configFile 必须用绝对路径,不能使用相对路径。

以采用 inline 方式为例,只需要在 App.config 或者 Web.Config 中配置 log4net 节点如下:

```
<log4net>
  <appender name="FileAppender" type="log4net.Appender.
FileAppender">
    <file value="log.txt" />
    <appendToFile value="true" />
    <layout type="log4net.Layout.SimpleLayout" />
  </appender>
</log4net>
<root>
  <level value="ALL" />
  <appender-ref ref="FileAppender" />
</root>
</log4net>
```

## 5 实例程序

官网上新加了一个例子程序 ConsoleApplication1, 可以下载来学习。为了使大家一步一步更明白项目建立过程,我们新建一个例子程序。该例子程序功能为:通过网页输入查询参数,程序通过 iBatis 查询 Oracle 数据库信息,并把结果返回显示在页面上。

## 5.1 建立项目

打开 Visual Studio 2012, 新建一个 ASP.NET Empty Web Application 项目 iBatisTest, Framework 框架选 4.0。

## 5.2 添加引用

通过官方网站下载 "IBatis.DataMapper.1.6.2.bin.zip" 并解压, 在项目中添加下面 dll 的引用: IBatisNet.Common.dll 和 IBatisNet.DataMapper.dll。因为要访问 Oracle 数据库,还需要在项目中引入 System.Data.OracleClient.dll。

## 5.3 添加 IBatis 环境配置文件

把解压目录下的 providers.config 和 sample.SqlMap.config 拷贝到项目根目录, 并把 sample.SqlMap.config 改名为 SqlMap.config, 把两个文件加入项目。本例子程序采用 "System.Data.OracleClient" 连接 Oracle 数据库, 打开 providers.config, 找到名称为 "oracleClient1.0", 修改 enabled 参数为 " true" 实启用该数据库驱动配置。

## 5.4 建立表

在 Oracle 数据库中建立 xxzx.post 表, 脚本如下:

```
CREATE TABLE XXZX.POST
(
  ID NUMBER(10,0),
  USERNAME VARCHAR2(100),
  POSTTIME DATE
```





);

插入测试数据, insert into XXZX.POST (id,username, posttime) values (10,'管理员',sysdate);

## 5.5 添加 XML 映射描述文件

向项目新增 XML 文件 Post.xml, 修改 XML 文件内容, 如图 3 所示。

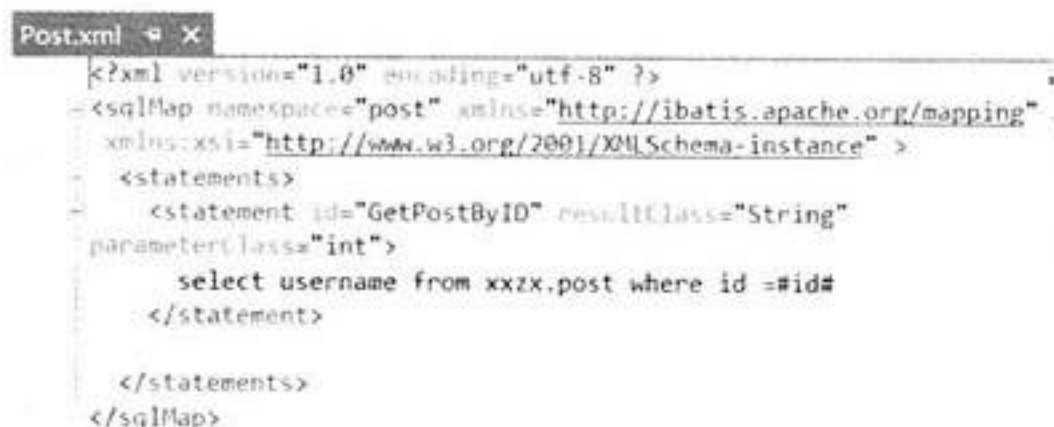


图 3 XML 映射文件

## 5.6 编写调用程序

向项目新增 Web Form 文件 default.aspx, 在页面添加控件, 如图 4 所示。

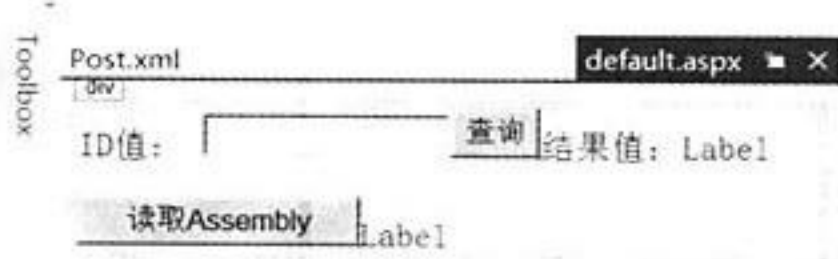


图 4 Web 页面

在 default.aspx.cs 增加 IBatisNet.DataMapper 的 using 应用。增加查询按钮控件事件, 代码如下:

```

protected void Button1_Click (object sender,
EventArgs e)
{
    ISqlMapper mapper = Mapper.Instance();//得到
    //ISqlMapper 实例
    int para = Convert.ToInt32(TextBox1.Text);
    string str = mapper.QueryForObject <string>("
    GetPostByID", para); //调用 QueryForObject 方法
    Label1.Text = str;
}
    
```

编译项目, 特别注意, 因为是 64 位操作系统下, 安装的 Oracle 客户端也是 64 位, 需要修改调试环境, 采用操作系统 IIS 来调试。具体操作为, 打开项目属性栏, 选择 Web 选项卡, 去掉 Use IIS Express 选项。Project Url 栏输入 http://localhost/iBatisTest。

运行项目时还是会提示数据库驱动程序不对, 这是因为安装的 System.Data.OracleClient.dll 版本和 providers.config 中记录

的版本不一致, 通过 Assembly 读取 System.Data.OracleClient.dll 文件 FullName, 并把该值填入 providers.config 文件 oracleClient1.0 驱动的 assemblyName 参数。读取代码如下:

```

Label2.Text = System.Reflection.Assembly.LoadFile(@"
E:\Ibatis.Net\ 文 章 \IbatisTest\IbatisTest\System.Data.
OracleClient.dll").FullName;
    
```

运行程序, 在 ID 值框里面输入 "10", 点击查询按钮, 就会返回正确的查询值, 结果如图 5 所示。



图 5 程序运行界面

## 5.7 增加日志记录

加入 IBatisNet.Common.Logging.Log4Net.dll 引用, 根据 IBatis 日志处理机制, 在 web.config 文件里面注册 IBatis 日志处理节点和 log4net 节点, 配置 IBatis 节点和 log4net, 注意注册 IBatis 日志处理节点和 log4net 节点需放在根 <configuration> 元素的第一个子元素。

再次运行并执行程序, 在根目录就会生成 log.txt 文件, 文件里面记录的 IBatis 运行是调试信息, 内容如图 6 所示。

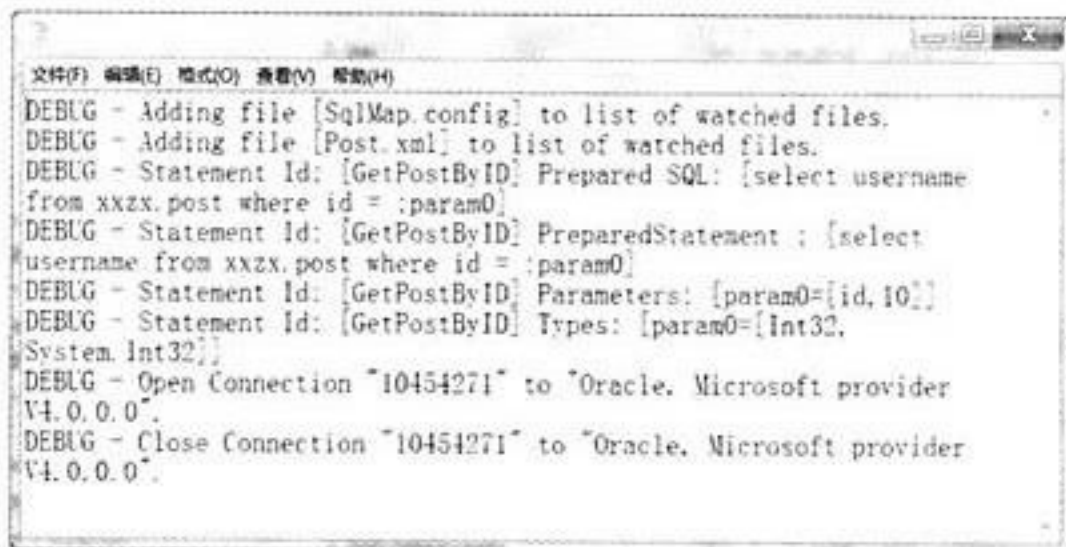


图 6 日志内容

## 6 结语

实例程序在 Windows7 (64 位) +VS2012+Oracle 11g (64 位) 上测试通过。通过实例程序可以看出, 只要配置好 XML 映射文件, 程序调用时只需要简单的几行代码就可以实现对数据库的查询操作, 大大的简化了对 ADO.NET 的操作。IBatis 便捷和高效。

(收稿日期: 2012-09-28)



# 基于 LabVIEW 的声卡双通道数字存储示波器

马海瑞 韩云东 郑振宇

**摘要:** 借助声卡的 A/D 功能设计了双通道数字存储示波器, 充分利用 LabVIEW 程序模块化等优势, 实现了对音频信号的测量、存储和回放, 具有性价比高、功能灵活等特点。程序设计方法可移植应用于数据采集编程。

**关键词:** 示波器; 双通道; LabVIEW 环境; 声卡; 存储

## 1 引言

声卡价格低廉、兼容性好、灵活通用, 而且其 DMA 数据传输方式极大地降低了 CPU 占用率, 在高速传送数据的同时可以让空闲的 CPU 执行数据转换、测量分析等其他任务, 能有效防止潜在的数据丢失和内存溢出错误。即使是主板集成的声卡, 其 16 位的 A/D 转换精度也能胜任许多工程测量和科学实验中的音频域信号采集工作。在 LabVIEW 中对声卡编程, 借助文件 I/O 等函数和属性节点, 实现了声卡双通道数字存储示波器 (以下简称声卡示波器), 其运行效果如图 1 所示。

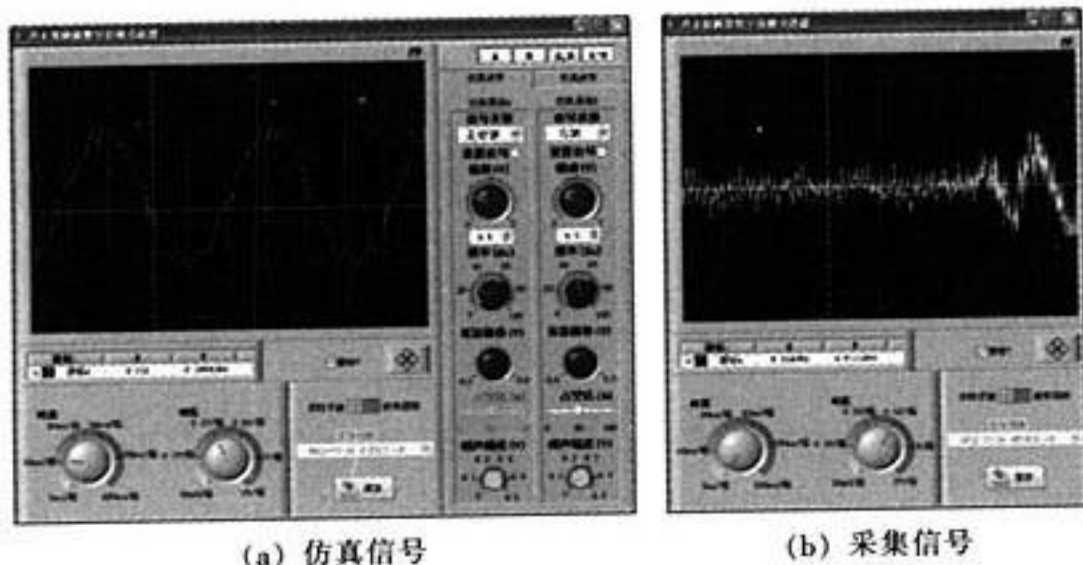


图 1 声卡示波器运行效果

## 2 波形采集

用 LabVIEW 调用声卡采集的基本流程如图 2 所示。声音函数在程序框图的数据流向上依次使用任务 ID, 以特定的方式对声音数据进行操作。Windows 操作系统中必须安装 DirectX 8.0 或更高版本才能使用声音函数。

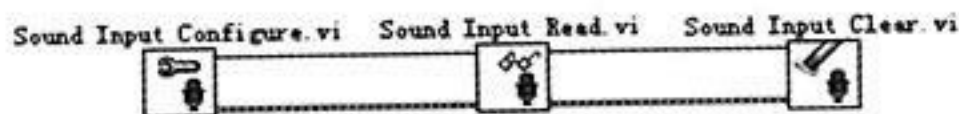


图 2 声卡采集基本流程

### 2.1 配置声音输入

Sound Input Configure 函数用于配置声音输入设备, 设置

采集率、通道数量和每个采样的位数, 触发采集后发送数据至缓存, 并返回引用当前声音任务的 ID。其中采样率参数一般至少有 8000Hz、11025Hz、22050Hz 和 44100Hz 等几个典型值, 应根据声卡具体情况和信号质量要求进行选择。

### 2.2 读取声音输入

声音输入操作利用计算机内存在 LabVIEW 和声音设备间开辟了一个缓冲区, 以确保声音数据传输通畅。Sound Input Read 函数用于从声音输入设备读取数据并将其存储为波形数组。数组中的一个波形即代表一个特定的通道, 因而双声道 (立体声) 的数据结构是包含了两个波形元素的数组。波形包含的是脉码调制 (PCM) 数据, 其中每个元素的大小与信号幅值成正比。在连续的信号采集中, 该函数应置于循环内。

### 2.3 声音输入清零

Sound Input Clear 函数用于关闭已经打开的声音输入设备, 释放声音输入操作所占用的一切资源, 以便后续或其他程序使用该设备。必要时, 还可借助“停止声音输入采集”和“启动声音输入采集”函数实现数据采集过程的中止和重新开始, 但此间并未释放相关软硬件资源。

## 3 波形存储与回放

### 3.1 波形存储

状态开关切换至实时示波状态时, 利用波形存储模块可将当前屏幕显示的波形保存为文件, 以便作进一步的分析。为便于回放并保持与其他 LabVIEW 文件的兼容性, 此外采用 LabVIEW 专用波形文件格式。通过点击前面板“保存”按钮控制“写入波形至文件”函数将当前感兴趣的波形写入硬盘, 代码如图 3 所示。



图 3 波形文件存储代码



“写入波形至文件”属高级文件函数，对文件的创建、打开、写入和关闭均在内部自动完成，简化了繁琐的编程流程。且波形文件类型本质上为二进制的记录文件，存储过程中只需进行少量处理，因而其读写速度更快。

为便于查看并防止文件重名，文件以存储时刻的日期时间（精确到秒）命名，其代码如图4所示。

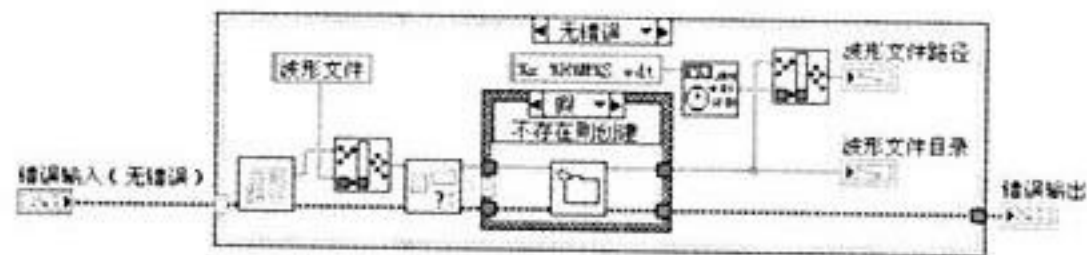


图4 波形文件路径及命名子VI代码

## 3.2 波形回放

将状态开关切换至波形回放状态则可调用以前保存的数据文件并显示其波形。数据文件名通过“罗列文件夹”函数送入下拉列表控件，可在示波器屏幕中有选择地回放，代码如图5所示。

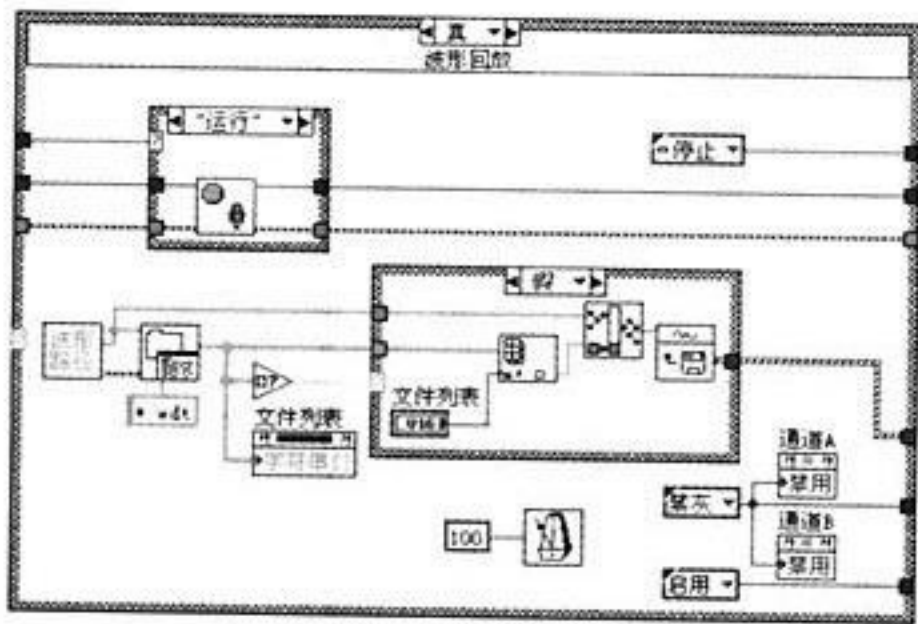


图5 波形回放代码

## 4 波形仿真

除了外部信号，声卡示波器在实时示波状态还有2路内部仿真信号源，用于生成正弦、三角、锯齿和方波等常用测试信号。单击“信号来源”选项卡可使波形图控件的数据源在仿真波形和采集波形之间切换。程序代码如图6所示。

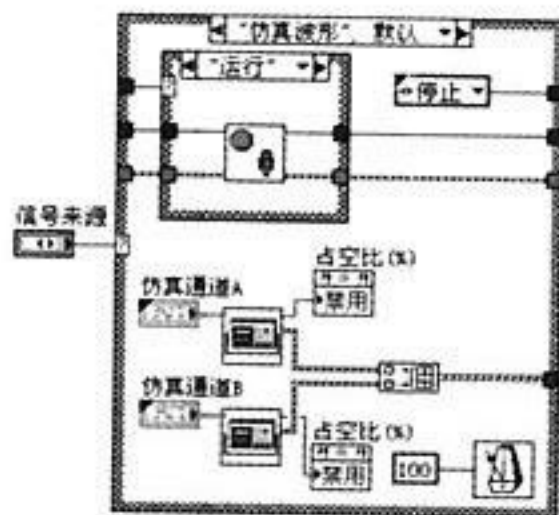


图6 仿真波形子框图

为充分发挥 LabVIEW 程序结构模块化的优势，将波形仿真生成等功能相对独立的模块创建为子VI。子VI未被调用时执行系统可取回它所占用的数据内存，不但能提高内存使用效率，还有利于后续开发和升级过程中的代码重用，而且层次化的VI组织也更易于维护和阅读。需要注意的是，子VI在主VI中被多处并行调用时，须在子VI的“执行”属性窗口中将其设置为重入执行，这样使每个被调用的VI实例都保持各自的信息状态，避免数据竞争。该子VI主要通过“基本函数发生器”等函数实现，其幅值、频率等信号参数可通过前面板控件调节，其主体代码如图7所示。

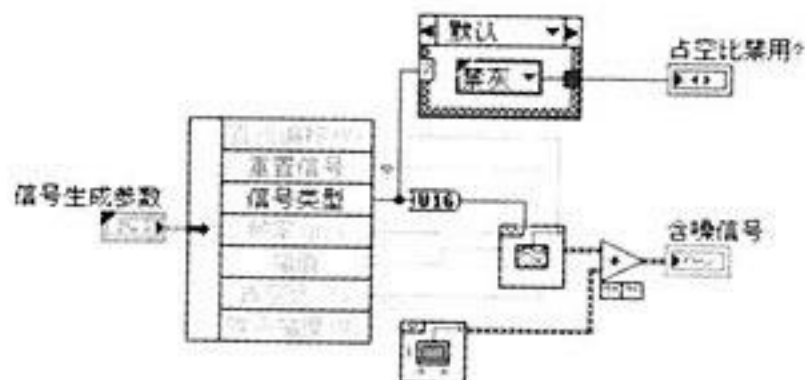


图7 波形仿真生成子VI代码

## 5 波形显示

### 5.1 档位调节

声卡示波器利用波形图控件显示信号波形。通过设置整型旋钮控件的文本标签来区别时基和幅基的不同档位，在调整时基和幅基旋钮档位过程中，为保持波形图控件中的横纵向网格数固定，程序利用波形图控件的标尺范围增量、最小值、最大值等属性节点对X、Y坐标刻度进行成倍数关系的控制，达到与传统示波器屏幕一致的效果，代码如图8所示。

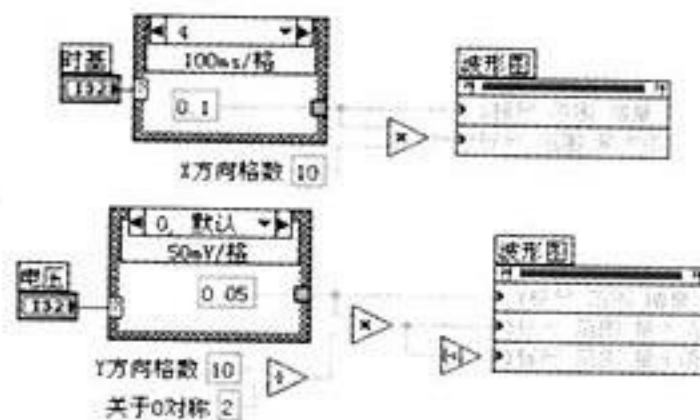


图8 波形图坐标档位控制代码

除了使用函数节点输出的结果，波形图控件的游标图例附件也提供了很好的辅助测量功能，例如，将游标关联至某通道的信号曲线上，通过移动游标线可以方便地读出曲线上任意点的坐标值，虽然读数精度受到像素和屏幕分辨率的限制，但用于定性分析或教学演示等精度要求不高的场合却很直观方便。

### 5.2 通道选择

声卡2声道各有一个模数转换器，它们在同一时钟下工作。这保证了2个通道同时被采样，使噪声的影响减到最小。



## PROGRAM LANGUAGE

双通道可同时工作，也可单独工作。其中，A-B 提供对一个信号的差分测量功能，即用声卡左右 2 个声道分别测量信号两端对地的电压后求其差值。需要注意的是，信号源两端各自的对地电压必须在指定的范围内，否则测量结果会因信号饱和而失去准确性。通道切换由条件结构不同分支中的数组操作函数实现，如图 9 所示。

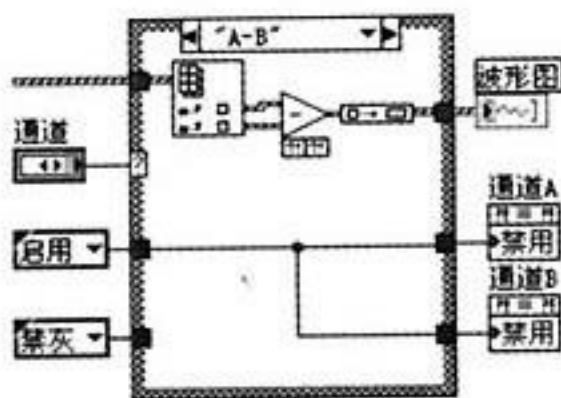


图 9 通道切换代码

## 6 结语

声卡双通道数字存储示波器充分利用了计算机的软硬件资

源，较好地实现了信号的时域测量功能，具有运行稳定、性价比高、使用方便等特点，给教学和科研带来很多方便。如果采用笔记本电脑并进一步结合 FFT 等波形测量函数，无需增加额外的硬件成本即可构成便携式频谱分析仪等功能更强大的测量系统。且声卡的工作流程与一般数据采集卡的操作过程类似，具有较好的普适性。

## 参考文献

- [1] National Instruments Corporation. LabVIEW 帮助 [M]. Texas: National Instruments Corporation, 2012.
- [2] 马海瑞, 田树森, 周爱军. 基于声卡的 LabVIEW 虚拟信号发生器设计 [J]. 国外电子测量技术, 2005, 24 (04): 27-29.
- [3] 澳德计算机控制技术有限公司. 澳德 DSoundStar 用户说明书 [M]. 福州: 澳德计算机控制技术有限公司, 2006.  
(收稿日期: 2012-11-07)

## 平台化发展 产业链共赢 UAP、CSP 彰显平台的力量

近日，亚太本土领先管理软件厂商用友软件股份有限公司（以下简称“用友”）正式发布 2013/新三年业务策略——加快平台化发展，推进产业链共赢；并向业界彰显平台化发展的最新成果——针对大中型企业的私有云平台 UAP 与 NC6 解决方案，针对小微企业的公有云平台 CSP 与四朵云服务。工业和信息化部、中国软件行业协会、业界专家、研究机构、产业链伙伴、机构投资者、媒体等 100 多人共同见证了这一重要时刻。

过去两年，用友公司从两个层面推进云战略并取得了实质性进展：针对大型企业、智慧城市、智慧园区提供了私有云解决方案，针对小微等企业提供了财税云、协同云、营销云和支付云服务。在有效服务于众多客户后，2013 年，用友正式向业界发布其最新的平台技术成果。

“未来软件与服务市场将是平台之战。”用友公司董事长兼 CEO 王文京表示，“对于软件企业而言，不仅是推进软件产品的云化、服务化，更重要的是，通过开放平台，最大限度地聚合大量第三方应用服务资源，以多、快、好、省地满足客户大量的个性化需求。”

UAP 是用友公司旗舰产品线 NC 的支撑平台。从 2001 年起，经过 4 个阶段的演变，历经 6 个版本，通过 8000 多家大中型企业客户的成功实践，UAP 已演进为全球技术领先的私有云平台：它不仅能将企业中现有的和未来的应用系统进行有机整合，形成一个开放的一体化信息平台，还迈向

了云间集成——能实现企业与供应商、经销商、电子商务平台的整合。UAP 包含了两个关键产品，商业分析平台“BQ”和数据处理平台“AE”。BQ 能实现闭环实时的智能分析，帮助企业实现精准决策，快速响应商业变化；AE 能支持数据压缩、列式存储、内存计算，支持结构化和非结构化数据服务。

基于 UAP 平台的 NC6.0 大型企业管理与电子商务解决方案已有众多成功案例。以港华燃气集团为例，其业务覆盖国内 21 个省/直辖市，目前有 103 个燃气项目，每年还以 8 到 10 个项目的速度增长，拥有 1 万多名员工，是国内领先的城市燃气供应商。自采用基于 UAP 的 NC6.0 私有云解决方案、一起进行信息系统的升级创新后，通过搭建港华燃气自身的私有云平台，可节约基础架构建设费用 90%，减少系统维护费 50%，未来将同时满足业务快速扩张和风险管控的目标管理需求。

CSP 是用友集团开发的公有云平台。目前，基于 CSP 公有云平台，用友集团旗下子公司畅捷通信息技术股份有限公司针对小微企业，提供了四朵云服务：协同云—企业空间，主要服务于企业内部协同及上下游协同，自 2012 年 11 月底产品发布，一个月累计用户数 949 家；营销云—旺铺、码客，主要服务于企业营销，旺铺自 2011 年和中国电信合作以来，已发展了 45 万体验用户，20 多万收费用户；财税云—会计家园，提供财税知识及工具服务；支付云—畅捷支付，提供线上和线下的企业支付服务。





# VC2010 制作单词、双词同现的频率统计工具

马创新

**摘要:** 介绍了使用 VC++ 2010 制作词语和词性频率统计工具的方法, 使用该工具能够轻松地统计出“词语频率”、“词性频率”、“双词频率”和“双词性频率”。

**关键词:** VC 语言; 统计; 词语; 词性

## 1 引言

语言研究者在研究某部文献的语言特点时, 经常要对其做分词和词性标注的处理, 并且还要统计出各个词语、各类词性出现的频率及概率, 有时甚至要统计出“紧邻的两个词语”或者“紧邻的两类词性”在全文中出现的频率及概率, 所以设计出一个用法简单、执行速度快的词语和词性频率统计工具是很有必要。

## 2 功能及特点

“词语和词性频率统计工具”提供了统计“词语频率”、“词性频率”、“双词频率”和“双词性频率”4种功能, 该工具的界面如图1所示。使用该程序时, 先点击“编辑浏览控件”中的文件夹图标, 就会打开“文件选择对话框”, 选择已经做过“分词和词性标注”处理的文本文件(后缀名为txt), 然后就可以根据任务需要点击“词语频率”、“词性频率”、“双词频率”和“双词性频率”4个命令按钮之一, 程序就会自动运行并把统计结果显示在列表控件中, 还能把统计结果保存在一个新的文件中, 这个新文件的名称会在界面最下方的编辑框中显示出来。



图1 词语和词性频率统计工具的界面

## 3 编程实现

### 3.1 新建项目

启动 Visual Studio 2010, 在“文件”菜单上, 指向“新

建”, 然后单击“项目”打开“新建项目”对话框。从 Visual C++ 项目列表中选择“MFC 应用程序”, 在“名称”框中键入“tongji”, 然后单击“确定”按钮。在“MFC 应用程序向导”的“应用程序类型”中, 选择“基于对话框”, 然后多次点击“下一步”, 最后点击“完成”即可。本项目的字符集要设置为“使用 Unicode 字符集”, 这项设置可在“项目→属性→配置属性→常规→字符集”中找到。

### 3.2 在对话框中放置控件

在对话框中放置如图1所示的各个控件, 在表1中, 按照从左到右, 从上到下的顺序对这些控件的类型、ID、标题、相关变量、作用等作了具体的介绍。

表1 控件设置

控件类型	ID	控件标题	相关变量	作用
静态文本	IDC_STATIC1	选择文件:		提示
编辑浏览控件	IDC_MFCEDITBROWSE1		m_fileName (CString 类型)	打开文件选择对话框, 选择文件
列表控件	IDC_LIST1		m_listWord (CListCtrl 类型)	列表显示程序处理的结果
命令按钮	IDC_BUTTON1	词语频率	m_wordFreq (CButton 类型)	统计词语频率的执行按钮
命令按钮	IDC_BUTTON2	词性频率	m_tagFreq (CButton 类型)	统计词性频率的执行按钮
命令按钮	IDC_BUTTON3	双词频率	m_twoWordFreq (CButton 类型)	统计双词频率的执行按钮
命令按钮	IDC_BUTTON4	双词性频率	m_twoTagFreq (CButton 类型)	统计双词性频率的执行按钮
静态文本	IDC_STATIC2	生成的文件:		提示
编辑控件	IDC_EDIT1		m_resultFile (CString 类型)	显示生成的文件名称

另外, 为了得到预想的显示效果, 要修改 ID 为 IDC\_LIST1 列表控件的外观属性, 把它的显示属性 (view) 由 Icon 类型改为 Report 类型。ID 为 IDC\_MFCEDITBROWSE1 编辑浏览控件的浏览模式 (Browse Mode) 要设置为文件浏览,



## PROGRAM LANGUAGE

其他控件均使用默认设置。

### 3.3 编写用于调用的函数代码

(1) 函数 Ansi2Unicode; 功能是把 ANSI 编码的多字节字符串转化为 Unicode 编码的宽字符串。计算机中现有的文本文件大部分是以 ANSI 字符集编码存储的, 所以在编程时要按照以多字节字符编码读入, 处理之后再转化为宽字节字符编码显示在列表控件中。

```
wstring Ansi2Unicode(string astrsrc)
{
    int nLength = ::MultiByteToWideChar ( CP_ACP, 0,
    astrsrc.c_str(), -1, NULL, 0);
    if(nLength <= 0) return wstring(L "");
    wchar_t *szbuffer = new wchar_t[nLength + 2];
    ::MultiByteToWideChar (CP_ACP, 0, astrsrc.c_str(), -1,
    szbuffer, nLength);
    wstring strnew = szbuffer;
    delete [] szbuffer;
    return strnew;
}
```

(2) 编写一个函数 changeFileName, 用于改变文件名称, 它生成的新文件名称是由旧文件名称加上计算机当前的时间所组成。

```
CString changeFileName( const CString& oldFileName )
{
    time_t t;
    time(&t);
    CString sTime( _tctime(&t) );
    sTime.Replace( ' ', '_' );
    sTime.Replace( ':', '-' );
    sTime = sTime.Mid( 4, 15 );
    CString fileName = oldFileName ;
    fileName = fileName.Mid(0, fileName.ReverseFind('.') );
    CString newFileName = fileName + "(" + sTime + ").txt";
    return newFileName;
}
```

### 3.4 程序运行的初始设置

在消息处理函数 BOOL CtongjiDlg::OnInitDialog () 的 “// TODO: 在此添加额外的初始化代码” 之后添加下面一段代码, 用于对列表控件做初始设置:

```
//设置列表视图扩展风格
m_listWord.SetExtendedStyle (LVS_EX_GRIDLINES |
LVS_EX_FULLROWSELECT|LVS_EX_ONECLICKACTIVATE);
//向列表视图中添加列
m_listWord.InsertColumn (0,_T (" 序号"),LVCFMT_LEFT,
100);
m_listWord.InsertColumn (1,_T (" 词语 或 词性 "),
LVCFMT_LEFT,200);
m_listWord.InsertColumn (2,_T (" 频率"),LVCFMT_LEFT,
120);
```

```
m_listWord.InsertColumn (3,_T (" 总数"),LVCFMT_LEFT,
100);
m_listWord.InsertColumn (4,_T (" 概率"),LVCFMT_LEFT,
100);
```

另外, 在 tongjiDlg.cpp 中要添加头文件:

```
#include<fstream>
#include<map>
#include<string>
using namespace std;
```

### 3.5 添加各个命令按钮的单击响应消息

(1) 标题为 “词语频率” 命令按钮的单击响应消息:

```
UpdateData(TRUE);
if(m_fileName == _T("") )
    return;
m_listWord.DeleteAllItems(); //删除列表控件中的原有
//内容
m_resultFile = _T ("...The program is running, please
wait a moment...");
UpdateData(false); UpdateWindow();

CString FileName = changeFileName( m_fileName );
//生成一个新文件名,用于输出统计结果
ifstream in; ofstream out;
in.open( m_fileName , ios::in);
if(! in) { AfxMessageBox(_T("Cannot open the file! "));
return; }
out.open( FileName, ios::out );
if(! out) { AfxMessageBox(_T("Cannot open the file! "));
return; }
map<string, unsigned> lex; //定义 map 容器,用于存
//放词语及其频率
map<string, unsigned>::iterator it;
pair<string,unsigned> next;
string word;
while(in >> word) //统计词频
{
    it = lex.find(word);
    if( it != lex.end() )
        it->second++;
    else
    {
        next = make_pair(word, 1);
        lex.insert(it, next );
    }
}
int total = 0;
for(it=lex.begin(); it!=lex.end(); ++it)
{
    out << it->first << " " << it->second << endl; //把统计
//结果输出到文件
    total = total + it->second; //计算所有词语的总频率
```





```

}
in.close(); out.close();
m_resultFile= FileName; //把新文件名显示在编辑框中

CStringW cstr; wstring wstr;
unsigned lines = 0; double prop;
for(it=lex.begin(); it!=lex.end(); ++it) //把统计结果显示在
//列表控件中
{
    cstr.Format(_T("%i"),lines+1);
    m_listWord.InsertItem(lines, cstr,lines); //显示序号
    wstr = Ascii2Unicode( it->first); //把 Ansi 编码的
//字符串转化为 Unicode 编码
    cstr = wstr.c_str();
    m_listWord.SetItemText(lines,1, cstr ); //显示词语
    cstr.Format(_T("%i"),it->second);
    m_listWord.SetItemText(lines,2,cstr); //显示当前词
//语的频率
    cstr.Format(_T("%i"),total);
    m_listWord.SetItemText(lines ,3 , cstr); //显示所
//有词语的总频率
    prop = (it->second * 1.0 ) / (total * 1.0); //计算当
//前词语的出现概率
    cstr.Format(_T("%lf"),prop);
    m_listWord.SetItemText(lines,4 , cstr); //显示当前
//词语的出现概率
    lines++;
}
lex.clear(); //清空容器
UpdateData(false); UpdateWindow();
return;

(2) 标题为“词性频率” 命令按钮的单击响应消息:
UpdateData(TRUE);
if(m_fileName ==_T(""))
    return;
m_listWord.DeleteAllItems();
m_resultFile = _T ("...The program is running, please wait
a moment...");
UpdateData(false); UpdateWindow();

CString FileName = changeFileName( m_fileName );
ifstream in; ofstream out;
in.open( m_fileName , ios::in);
if(! in) { AfxMessageBox(_T("Cannot open the file! ") );
return; }
out.open( FileName, ios::out );
if(! out) { AfxMessageBox(_T("Cannot open the file! ") );
return; }
map<string, unsigned> lex;
map<string, unsigned>::iterator it;
string word, tag;

```

```

pair<string,unsigned> next;
while(in >> word)
{
    if ( word.find('/') < 0 || word.find('/') > ( word.size() - 1
) ) //寻找词性标记
        continue;
    tag = word.substr( word.find('/') ); //从字符串中切出
//词性标记
    it = lex.find(tag);
    if( it !=lex.end() ) //统计各种词性的出现频率
        it->second++;
    else
    {
        next = make_pair(tag, 1);
        lex.insert(it, next );
    }
}

int total = 0;
for(it=lex.begin(); it!=lex.end(); ++it)
{
    out << it->first << " " << it->second << endl;
//把统计结果输出到文件
    total = total + it->second; //计算出总频率
}
in.close(); out.close();
m_resultFile= FileName;

CStringW cstr; wstring wstr;
unsigned lines = 0; double prop;
for(it=lex.begin(); it!=lex.end(); ++it) //把统计结果显示在
//列表控件中
{
    cstr.Format(_T("%i"),lines+1);
    m_listWord.InsertItem(lines, cstr,lines); //显示序号
    wstr = Ascii2Unicode( it->first); //把 Ansi 编码的
//字符串转化为 Unicode 编码
    cstr = wstr.c_str();
    m_listWord.SetItemText(lines,1, cstr ); //显示词性标记
    cstr.Format(_T("%i"),it->second);
    m_listWord.SetItemText(lines,2,cstr); //显示当前词
//性的出现频率
    cstr.Format(_T("%i"),total);
    m_listWord.SetItemText(lines ,3 , cstr); //显示总频率
    prop = (it->second * 1.0 ) / (total * 1.0); //计算当
//前词性的出现概率
    cstr.Format(_T("%lf"),prop);
    m_listWord.SetItemText(lines,4 , cstr); //显示当
//前词性的概率
    lines++;
}

```





## PROGRAM LANGUAGE

```
lex.clear();
UpdateData(false); UpdateWindow();
return;
(3) 标题为“双词频率”命令按钮的单击响应消息:
UpdateData(TRUE);
if(m_fileName == _T(""))
    return;
m_listWord.DeleteAllItems();
m_resultFile = _T ("...The program is running, please
wait a moment...");
UpdateData(false); UpdateWindow();

CString FileName = changeFileName( m_fileName );
ifstream in;    ofstream out;
in.open( m_fileName , ios::in);
if(! in) { AfxMessageBox(_T("Cannot open the file! "));
return; }
out.open( FileName, ios::out );
if(! out) { AfxMessageBox(_T("Cannot open the file! "));
return; }

map<string, unsigned> lex;
map<string, unsigned>::iterator it;
pair<string, unsigned> next;
string wordOne = "", wordTwo = "", binWords = "";
in >> wordOne;
while(in >> wordTwo )
{
    binWords = wordOne + " " + wordTwo;
    it = lex.find( binWords );
    if( it != lex.end() ) //统计“紧邻双词”的出现频率
        it->second++;
    else
    {
        next = make_pair(binWords, 1);
        lex.insert(it, next );
    }

    if( in >> wordOne )
    {
        binWords = wordTwo + " " + wordOne ;
        it = lex.find( binWords );
        if( it != lex.end() ) //统计“紧邻双词”的出现频率
            it->second++;
        else
        {
            next = make_pair(binWords, 1);
            lex.insert(it, next );
        }
    }
} //end while
```

```
int total = 0;
for(it=lex.begin(); it!=lex.end(); ++it)
{
    out << it->first << " " << it->second << endl; //输出到
//文件
    total = total + it->second; //计算总频率
}
in.close(); out.close();
m_resultFile= FileName;

CStringW cstr;    wstring wstr;
unsigned lines = 0; double prop;
for(it=lex.begin(); it!=lex.end(); ++it) //把统计结果显示
//在列表控件中
{
    cstr.Format(_T("%i"),lines+1);
    m_listWord.InsertItem(lines, cstr,lines); //显示序号
    wstr = Ascii2Unicode( it->first); //把 Ansi 编码
//的字符串转化为 Unicode 编码
    cstr = wstr.c_str();
    m_listWord.SetItemText(lines,1, cstr ); //显示“紧邻
//的双词”
    cstr.Format(_T("%i"),it->second);
    m_listWord.SetItemText(lines,2,cstr); //显示频率
    cstr.Format(_T("%i"),total);
    m_listWord.SetItemText(lines ,3 , cstr); //显示总频率
    prop = (it->second * 1.0 ) / (total * 1.0); //计算概率
    cstr.Format(_T("%lf"),prop);
    m_listWord.SetItemText(lines,4 , cstr); //显示概率
    lines++;
}
lex.clear();
UpdateData(false); UpdateWindow();
return;
(4) 标题为“双词性频率”命令按钮的单击响应消息:
UpdateData(TRUE);
if(m_fileName == _T(""))
    return;
m_listWord.DeleteAllItems();
m_resultFile = _T ("...The program is running, please wait
a moment...");
UpdateData(false); UpdateWindow();
CString FileName = changeFileName( m_fileName );
ifstream in;    ofstream out;
in.open( m_fileName , ios::in);
if(! in) { AfxMessageBox(_T("Cannot open the file! "));
return; }
out.open( FileName, ios::out );
if(! out) { AfxMessageBox(_T("Cannot open the file! "));
return; }
map<string, unsigned> lex;
```





```

map<string, unsigned>::iterator it;
pair<string, unsigned> next;
string wordOne = "", wordTwo = "", binTags = "";
in >> wordOne;
while(in >> wordTwo)
{
    if ( ( wordOne.find('/') >= 0 && wordOne.find('/') <
wordOne.size() ) && ( wordTwo.find('/') >= 0 && wordTwo.
find('/') < wordTwo.size() ) )
    {
        binTags = wordOne.substr( wordOne.find('/') ) +
" " + wordTwo.substr(wordTwo.find('/'));
        //从紧邻的两个词语中切出两个词性标记
        it = lex.find( binTags );
        if( it != lex.end() ) //统计“紧邻两个词性”的出现频率
            it->second++;
        else
        {
            next = make_pair(binTags, 1);
            lex.insert(it, next );
        }
    }
    if( in >> wordOne && ( wordOne.find('/') >= 0 && wordOne.
find('/') < wordOne.size() ) && ( wordTwo.find('/') >= 0 &&
wordTwo.find('/') < wordTwo.size() ) )
    {
        binTags = wordTwo.substr(wordTwo.find('/')) +
" "+wordOne.substr( wordOne.find('/'));
        //从紧邻的两个词语中切出两个词性标记
        it = lex.find(binTags);
        if( it != lex.end() ) //统计“紧邻两个词性”的出现频率
            it->second++;
        else
        {
            next = make_pair(binTags, 1);
            lex.insert(it, next );
        }
    }
} //end while

int total = 0;
for(it=lex.begin(); it != lex.end(); ++it)
{
    out << it->first << " " << it->second << endl;
//把统计结果输出到文件
    total = total + it->second;    //计算出总频率
}
in.close(); out.close();
m_resultFile= FileName;

CStringW cstr;    wstring wstr;

```

```

unsigned lines = 0; double prop;
for(it=lex.begin(); it != lex.end(); ++it) //把统计结果显示
//在列表控件中
{
    cstr.Format(_T("%i"),lines+1);
    m_listWord.InsertItem(lines, cstr,lines);    //显示序号
    wstr = Ascii2Unicode( it->first);    //把 Ansi 编码
//的字符串转化为 Unicode 编码
    cstr = wstr.c_str();
    m_listWord.SetItemText(lines,1, cstr ); //显示“双词性”
    cstr.Format(_T("%i"),it->second);
    m_listWord.SetItemText(lines,2,cstr);    //显示频率
    cstr.Format(_T("%i"),total);
    m_listWord.SetItemText(lines ,3 , cstr);    //显示总频率
    prop = (it->second * 1.0 ) / (total * 1.0); //计算概率
    cstr.Format(_T("%lf"),prop);
    m_listWord.SetItemText(lines,4 , cstr);    //显示概率
    lines++;
}
lex.clear();
UpdateData(false);    UpdateWindow();
return;

```

## 4 结语

介绍了使用 VC++ 2010 制作词语和词性频率统计工具的方法，凭借该工具，能够轻松地统计出已经做过“分词和词性标注”文件的“词语频率”、“词性频率”、“双词频率”和“双词性频率”。所有代码均在 VC++2010 下通过测试，具有较强的可移植性。

## 参考文献

- [1] D. S. Malik, 著, 晏海华, 等, 译. C++编程—数据结构与程序设计方法 [M]. 北京: 电子工业出版社, 2003.
  - [2] 马创新. WEB 文献资料采集系统. 计算机系统应用, 2012, 21 (7): 9-12,37.
  - [3] 梁水, 王家胜. 视频学 Visual C++ [M]. 北京: 人民邮电出版社, 2009.
  - [4] 马创新, 陈小荷, 曲维光, 陆鹏飞. 《论语》与其注疏文献对齐语料库的构建. 现代教育技术, 2012, (7).
- (收稿日期: 2012-11-12)



# Java 开发目录同步程序

江 洪

**摘 要:** 使用 Java 语言开发了目录同步程序。程序分为服务端和客户端。服务端设置和实时扫描要进行同步的目录并通过 Java 套接字将读到的文件数据传给客户端,并由客户端写到相应目录下。

**关键词:** Java 语言; 目录同步; 套接字

## 1 引言

Java 语言是一种十分流行的程序开发语言。由 SUN 公司推出,具有简易性、健壮性、安全性、可移植性等许多特性。可以用它开发各种应用程序,是目前主流的开发语言。这里使用 Java 语言开发一个实用的目录同步程序,通过实例讲解基本的 Java 程序开发技术。

## 2 开发思路

在网络环境中,往往存在这样的需求。网络中的一台或几台计算机需要和网络中的某一台计算机中的目录文件保持一致。也就要同步的目录名和文件名以及文件内容要完全一样。如果文件发生变化,变化后的文件将被实时更新。要实现这样的功能,可以设置有目录同步要求的计算机为服务器,要进行同步的计算机设置为客户端。双方通过套接字传递数据,完成同步的功能。

服务器端设置要进行同步的目录名,然后扫描该目录下的文件,将要同步的文件名、文件校验码、文件时间、文件大小信息传给客户端;客户端检查要同步的文件,如发现文件不一致,就进行文件同步,如果一致,就确认同步完成。

## 3 程序代码

程序中有如下类,用于搜索目录,并将搜索到的文件放到一个 ArrayList 中:

```
class searchdirthread extends Thread //搜索目录
{
    List <String>resultList=new ArrayList<String>();
    Socket connection;
    DataInputStream in=null;
    DataOutputStream out=null;
    String DIR1=null,DIR2=null,DIR3=null,DIR4=null,DIR5=null;
    String DIR6 =null,DIR7 =null,DIR8 =null,DIR9 =null,DIR10 =
    null;
```

```
TextArea text1;
```

```
public searchdirthread(Socket _connection,TextArea _text1,
    String _DIR1,String _DIR2,String _DIR3,
    String _DIR4,String _DIR5,String _DIR6,
    String _DIR7,String _DIR8,String _DIR9,
    String _DIR10)
{
    connection=_connection;
    text1=_text1;
    try
    {
        in=new DataInputStream(connection.getInputStream());
        out =new DataOutputStream (connection.
        getOutputStream());
    }
    catch(Exception e){}
    DIR1=_DIR1;DIR2=_DIR2;DIR3=_DIR3;DIR4=_DIR4;
    DIR5=_DIR5;DIR6=_DIR6;DIR7=_DIR7;DIR8=_DIR8;
    DIR9=_DIR9;DIR10=_DIR10;
    start();
}
public static void findFiles(String baseDirName,String
    targetFileName,
    List <String>fileList) //查找文件
{
    File baseDir=new File(baseDirName);
    String tempName=null,str=null;
    File tempFile;
    File[] files=baseDir.listFiles();

    if(! baseDir.exists()||! baseDir.isDirectory()) return;
    for(int i=0;i<files.length;i++)
    {
        tempFile=files[i];
        if(tempFile.isDirectory())
            findFiles(tempFile.getAbsolutePath(),
```





```

        targetFileName,fileList);
    else
        if(tempFile.isFile())
        {
            tempName=tempFile.getName();
            if(wildcardMatch(targetFileName,tempName))
            {
                str=tempFile.getAbsolutePath().toString();
                if(tempFile.length()>0) fileList.add(str);
            }
        }
    }
}
//匹配通配符
private static boolean wildcardMatch (String pattern,String
str)
{
    int patternLength=pattern.length();
    int strLength=str.length();
    int strIndex=0;
    char ch;

    for(
        int patternIndex =0;patternIndex <patternLength;
        patternIndex++)
    {
        ch=pattern.charAt(patternIndex);
        if(ch=='*')
        {
            while(strIndex<strLength)
            {
                if(wildcardMatch(pattern.substring(patternIndex+1),
                    str.substring(strIndex)))
                    return true;
                strIndex++;
            }
        }
        else
            if(ch=='?')
            {
                strIndex++;
                if (strIndex>strLength) return false;
            }
            else
            {
                if((strIndex>=strLength)||((ch!=str.charAt(strIndex))))
                    return false;
                strIndex++;
            }
    }
    return (strIndex==strLength);
}

```

```

public void run() //线程运行
{
    String s=null;
    File file;
    String md5str=null;
    int i;
    ServerThread handler =new ServerThread (connection,
    text1,resultList);

    while(true)
    {
        try
        {
            if(resultList.size()==0)
            {
                //查找文件,结果在 resultList
                if(! DIR1.equals("")) findFiles(DIR1,"",resultList);
                if(! DIR2.equals("")) findFiles(DIR2,"",resultList);
                if(! DIR3.equals("")) findFiles(DIR3,"",resultList);
                if(! DIR4.equals("")) findFiles(DIR4,"",resultList);
                if(! DIR5.equals("")) findFiles(DIR5,"",resultList);
                if(! DIR6.equals("")) findFiles(DIR6,"",resultList);
                if(! DIR7.equals("")) findFiles(DIR7,"",resultList);
                if(! DIR8.equals("")) findFiles(DIR8,"",resultList);
                if(! DIR9.equals("")) findFiles(DIR9,"",resultList);
                if(! DIR10.equals("")) findFiles(DIR10,"",resultList);
                if(resultList.size()>0)
                {
                    //发出第一个文件开始请求
                    s="FILE1*+";
                    s=s+resultList.get(0)+";
                    file=new File(resultList.get(0).toString());
                    try
                    {
                        md5str=MD5Util.getFileMD5String(file);
                    }
                    catch(Exception e){}
                    s=s+md5str+";
                    s=s+Long.toString(file.lastModified())+";
                    s=s+Long.toString(file.length())+";
                    out.write(s.getBytes(),0,s.getBytes().length);
                    out.flush();
                }
            }
            Thread.sleep(1);
        }
        catch(Exception e)
        {
            if(e.toString().equals(
                "java.net.SocketException: Connection reset by peer:
                socket write error"))
                break;
        }
    }
}

```





## PROGRAM LANGUAGE

```

    }
}
}
}

```

服务端和客户端按以下规则传递数据，如表 1 所示。

表 1 服务端与客户端数据传递规则

名称	格式	说明
文件开始请求	FILE1+分割符+文件全路径名+分割符+文件校验码+分割符+文件时间+分割符+文件大小+分割符	服务端向客户端发送该请求，表示请求同步该文件
文件开始已同步应答	FILE2_0+分割符+文件全路径名+分割符	客户端向服务端返回该应答，表示该文件已经同步过
文件开始应答	FILE2_1+分割符+文件全路径名+分割符	客户端向服务端返回该应答，表示同意同步该文件
文件数据请求	FILE3+分割符+文件全路径名+分割符+数据大小+分割符+数据	服务端向客户端发送该请求，表示传一块文件数据
文件数据应答	FILE4+分割符+文件全路径名+分割符	客户端向服务端返回该应答，表示该块数据接收成功
文件结束已同步请求	FILE5_0+分割符+文件全路径名+分割符+文件校验码+分割符+文件时间+分割符+文件大小+分割符	服务端向客户端发送该请求，表示结束该已同步过的文件
文件结束请求	FILE5_1+分割符+文件全路径名+分割符+文件校验码+分割符+文件时间+分割符+文件大小+分割符	服务端向客户端发送该请求，表示结束该文件
文件结束失败应答	FILE6_0+分割符+文件全路径名+分割符+文件校验码+分割符+文件时间+分割符+文件大小+分割符	客户端向服务端返回该应答，表示该文件同步失败
文件结束成功应答	FILE6_1+分割符+文件全路径名+分割符+文件校验码+分割符+文件时间+分割符+文件大小+分割符	客户端向服务端返回该应答，表示该文件同步成功
文件结束已同步应答	FILE6_2+分割符+文件全路径名+分割符+文件校验码+分割符+文件时间+分割符+文件大小+分割符	客户端向服务端返回该应答，表示该文件已同步完成

使用如下类，用于读写服务端套接字：

```

class ServerThread extends Thread //服务端读写 SOCKET
{
    Socket connection;
    DataInputStream in=null;
    DataOutputStream out=null;
    TextArea text1;
    List <String>resultList;

```

```

    public ServerThread(Socket _connection,TextArea _text1,
        List <String>_resultList)
    {
        connection=_connection;
        text1=_text1;
        resultList=_resultList;
        start();
        try
        {
            in=new DataInputStream(connection.getInputStream());
            out =new DataOutputStream (connection.
getOutputStream());
        }
        catch(Exception e){}
    }
    public void run()
    {
        int readbytes=0;
        byte buf[]=new byte[1600];
        String md5str=null;
        File file;
        String s,str=null,str1,s1;
        DataInputStream filein=null;
        byte buffer[] = new byte[1000];
        byte buffer1[] = new byte[1600];
        byte buffer2[] = new byte[500];
        int count,i,j,k=1;

        while(true)
        {
            try
            {
                readbytes=in.read(buf,0,1600);
                if(readbytes>0)
                {
                    s=new String(buf);
                    //文件结束应答
                    if(buf[0]=='F'&&buf[1]=='I'&&buf[2]=='L'&&buf[3]=='E'&&buf[4]=='6')
                    {
                        if(k<resultList.size())
                        {
                            //发文件开始请求
                            s1="FILE1"+k;
                            s1=s1+resultList.get(k)+"\n";
                            file=new File(resultList.get(k).toString());
                            try
                            {
                                md5str=MD5Util.getFileMD5String(file);
                            }
                            catch(Exception e){}

```





```

s1=s1+md5str+"|";
s1=s1+Long.toString(file.lastModified())+"|";
s1=s1+Long.toString(file.length())+"|";
out.write(s1.getBytes(),0,s1.getBytes().length);
out.flush();
k++;
}
else //本轮同步完成
{
    resultList.clear();k=1;
}
}
if(buf[0]=='F'&&buf[1]=='I'&&buf[2]=='L'&&buf[3]=='E'&&buf[4]=='6'&&
    buf[5]=='_'&&buf[6]=='0') //文件结束同步失
//败应答
{
    s=s.substring(8);
    s=s.substring(0,s.indexOf("|"));
    filein.close();
    InetAddress clientIP = connection.getInetAddress();

    str="向客户端"+clientIP.getHostAddress();
    str=str+"同步文件"+s+"失败";
    str1=text1.getText();
    str1=str1+str+"\n";
    text1.setText(str1);
    text1.setCaretPosition(str1.length());
}
if(buf[0]=='F'&&buf[1]=='I'&&buf[2]=='L'&&buf[3]=='E'&&buf[4]=='6'&&
    buf[5]=='_'&&buf[6]=='1') //文件结束同步成功
//应答
{
    s=s.substring(8);
    s=s.substring(0,s.indexOf("|"));
    filein.close();
    InetAddress clientIP = connection.getInetAddress();

    str="向客户端"+clientIP.getHostAddress();
    str=str+"同步文件"+s+"成功";
    str1=text1.getText();
    str1=str1+str+"\n";
    text1.setText(str1);
    text1.setCaretPosition(str1.length());
}
if(buf[0]=='F'&&buf[1]=='I'&&buf[2]=='L'&&buf[3]=='E'&&buf[4]=='6'&&
    buf[5]=='_'&&buf[6]=='2') //文件结束已同步应答
{
    s=s.substring(8);

```

```

s=s.substring(0,s.indexOf("|"));
}
//文件数据应答
if(buf[0]=='F'&&buf[1]=='I'&&buf[2]=='L'&&buf[3]=='E'&&buf[4]=='4')
{
    s=s.substring(6);
    s=s.substring(0,s.indexOf("|"));
    count=filein.read(buffer);
    if(count!=-1) //文件未结束
    {
        //发文件数据请求
        str="FILE3"+s+"|"+count+"|";
        buffer2=str.getBytes();
        for(i=1;i<=buffer2.length;i++) buffer1[i-1]=buffer2
[i-1];

        for(j=1;j<=count;j++) buffer1[i+j-2]=buffer[j-1];
        out.write(buffer1,0,buffer2.length+count);
        out.flush();
    }
    else //文件结束
    {
        //发文件结束请求
        str="FILE5_1"+s+"|";
        str=str+s+"|";
        file=new File(s);
        try
        {
            md5str=MD5Util.getFileMD5String(file);
        }
        catch(Exception e){}
        str=str+md5str+"|";
        str=str+Long.toString(file.lastModified())+"|";
        str=str+Long.toString(file.length())+"|";
        out.write(str.getBytes(),0,str.getBytes().length);
        out.flush();
    }
}
//文件开始已同步应答
if(buf[0]=='F'&&buf[1]=='I'&&buf[2]=='L'&&buf[3]=='E'&&buf[4]=='2'&&
    buf[5]=='_'&&buf[6]=='0')
{
    //发文件结束已同步请求
    s=s.substring(8);
    s=s.substring(0,s.indexOf("|"));
    str="FILE5_0"+s+"|";
    str=str+s+"|";
    file=new File(s);
    try
    {

```





## PROGRAM LANGUAGE

```

        md5str=MD5Util.getFileMD5String(file);
    }
    catch(Exception e){}
    str=str+md5str+"|";
    str=str+Long.toString(file.lastModified())+"|";
    str=str+Long.toString(file.length())+"|";
    out.write(str.getBytes(),0,str.getBytes().length);
    out.flush();
}
//文件开始应答
if(buf[0]=='F'&&buf[1]=='l'&&buf[2]=='L'&&buf[3]=='E'&&buf[4]=='2'&&
    buf[5]=='_'&&buf[6]=='1')
{
    //发文件数据请求
    s=s.substring(8);
    s=s.substring(0,s.indexOf("|"));
    filein=new DataInputStream (new FileInputStream
(s));
    count=filein.read(buffer);
    str="FILE3*|"+s+"|"+count+"|";
    buffer2=str.getBytes();
    for(i=1;i<=buffer2.length;i++) buffer1[i-1]=buffer2
[i-1];
    for(j=1;j<=count;j++) buffer1[i+j-2]=buffer[j-1];
    out.write(buffer1,0,buffer2.length+count);
    out.flush();
}
}
catch(Exception e)
{
    if(e.toString().equals(
        "java.net.SocketException: Connection reset")) break;
}
}
}
}

```

使用如下类，用于读写客户端套接字：

```

class clientdaemon extends Thread //客户端读写 SOCKET
{
    private Socket client;
    DataInputStream in=null;
    DataOutputStream out=null;
    TextArea text1;

    public clientdaemon(Socket _client,TextArea _text1)
    {
        client=_client;
        text1=_text1;
        start();
    }
}

```

```

try
{
    in=new DataInputStream(client.getInputStream());
    out=new DataOutputStream(client.getOutputStream());
}
catch(Exception e){}
}
public void run()
{
    int readbytes=0;
    byte buf[]=new byte[1600];
    String s,s1,s2,s3,str,str1;
    int flag,len,i;
    DataOutputStream fileout=null;
    int transferring=0;

    while(true)
    {
        try
        {
            readbytes=in.read(buf,0,1600);
            if(readbytes>0)
            {
                s=new String(buf);
                //文件结束请求
                if(buf[0]=='F'&&buf[1]=='l'&&buf[2]=='L'&&buf[3]=='E'&&buf[4]=='5'&&
                    buf[5]=='_'&&buf[6]=='1')
                {
                    transferring=0;
                    fileout.close();
                    s=s.substring(8);
                    s=s.substring(0,s.indexOf("|"));
                    File file=new File(s);
                    s1=new String(buf);
                    s1=s1.substring(s1.indexOf("|")+1);
                    s1=s1.substring(s1.indexOf("|")+1);
                    s1=s1.substring(0,s1.indexOf("|"));
                    s2=new String(buf);
                    s2=s2.substring(s2.indexOf("|")+1);
                    s2=s2.substring(s2.indexOf("|")+1);
                    s2=s2.substring(s2.indexOf("|")+1);
                    s2=s2.substring(0,s2.indexOf("|"));
                    s3=new String(buf);
                    s3=s3.substring(s3.indexOf("|")+1);
                    s3=s3.substring(s3.indexOf("|")+1);
                    s3=s3.substring(s3.indexOf("|")+1);
                    s3=s3.substring(s3.indexOf("|")+1);
                    s3=s3.substring(0,s3.indexOf("|"));
                    flag=1;
                    if(file.exists())

```





```

{
    file.setLastModified(Long.parseLong(s2));
    if(Long.parseLong(s2) != file.lastModified()) flag=0;
    if(Long.parseLong(s3) != file.length()) flag=0;
    if (! s1.equals (MD5Util.getFileMD5String
(file))) flag=0;
}
if(flag==1)
{
    //发文件结束同步成功应答
    str="FILE6_1"+"|"+s+"|";
    out.write(str.getBytes(),0,str.getBytes().length);
    out.flush();
    str="文件\\"+s+"\ 同步成功";
    str1=text1.getText();
    str1=str1+str+"\n";
    text1.setText(str1);
    text1.setCaretPosition(str1.length());
}
if(flag==0)
{
    //发文件结束同步失败应答
    str="FILE6_0"+"|"+s+"|";
    out.write(str.getBytes(),0,str.getBytes().length);
    out.flush();
    str="同步文件"+s+"失败";
    str1=text1.getText();
    str1=str1+str+"\n";
    text1.setText(str1);
    text1.setCaretPosition(str1.length());
}
}
//文件结束已同步请求
if(buf[0]=='F'&&buf[1]=='1'&&buf[2]=='L'&&buf[3]=='
'E'&&buf[4]=='5'&&
    buf[5]=='_'&&buf[6]=='0')
{
    //发文件结束已同步应答
    transferring=0;
    s=s.substring(8);
    s=s.substring(0,s.indexOf("|"));
    str="FILE6_2"+"|"+s+"|";
    out.write(str.getBytes(),0,str.getBytes().length);
    out.flush();
}
//文件数据请求
if(buf[0]=='F'&&buf[1]=='1'&&buf[2]=='L'&&buf[3]=='
'E'&&buf[4]=='3')
{
    s=s.substring(6);
    s=s.substring(0,s.indexOf("|"));

```

```

s1=new String(buf);
s1=s1.substring(s1.indexOf("|")+1);
s1=s1.substring(s1.indexOf("|")+1);
s1=s1.substring(0,s1.indexOf("|"));
len=Integer.parseInt(s1);
for(i=1;i<=len;i++)
    fileout.write(buf[i-1+6+s1.length()+1+s.getBytes().
length+1]);
//发文件数据应答
str="FILE4"+"|"+s+"|";
out.write(str.getBytes(),0,str.getBytes().length);
out.flush();
}
//文件开始请求
if(buf[0]=='F'&&buf[1]=='1'&&buf[2]=='L'&&buf[3]=='
'E'&&buf[4]=='1')
{
    if(transferring==1) continue;
    transferring=1;
    s=s.substring(6);
    s=s.substring(0,s.indexOf("|"));
    File file=new File(s);
    File parent=file.getParentFile();
    if(parent != null&&! parent.exists()) parent.mkdirs();
    s1=new String(buf);
    s1=s1.substring(s1.indexOf("|")+1);
    s1=s1.substring(s1.indexOf("|")+1);
    s1=s1.substring(0,s1.indexOf("|"));
    s2=new String(buf);
    s2=s2.substring(s2.indexOf("|")+1);
    s2=s2.substring(s2.indexOf("|")+1);
    s2=s2.substring(s2.indexOf("|")+1);
    s2=s2.substring(0,s2.indexOf("|"));
    s3=new String(buf);
    s3=s3.substring(s3.indexOf("|")+1);
    s3=s3.substring(s3.indexOf("|")+1);
    s3=s3.substring(s3.indexOf("|")+1);
    s3=s3.substring(s3.indexOf("|")+1);
    s3=s3.substring(0,s3.indexOf("|"));
    flag=1;
    if(file.exists())
        if(Long.parseLong(s3)==file.length())
            if(Long.parseLong(s2)==file.lastModified())
                if(s1.equals(MD5Util.getFileMD5String(file)))
                    flag=0;
    if(flag==0)
    {
        //发文件开始已同步应答
        str="FILE2_0"+"|"+s+"|";
        out.write(str.getBytes(),0,str.getBytes().length);
        out.flush();
    }
}

```





## PROGRAM LANGUAGE

```

    }
    if(flag==1)
    {
        //发文件开始应答
        file.delete();
        file.createNewFile();
        fileout =new DataOutputStream (new
FileOutputStream(s));
        str="FILE2_1"+" "+s+" ";
        out.write(str.getBytes(),0,str.getBytes().length);
        out.flush();
    }
}
}
}
catch(Exception e)
{
    transferring=0;
    if(e.toString().equals(
        "java.net.SocketException: Connection reset")) break;
}
}
}
}
}

```

使用如下类，用于进行文件校验，以保证文件可靠传输：

```

class MD5Util //MD5 校验
{
    protected static char hexDigits[]={ '0','1','2','3','4','5','6',
        '7','8','9','a','b','c','d','e','f' };
    protected static MessageDigest messagedigest=null;
    static
    {
        try
        {
            messagedigest=MessageDigest.getInstance("MD5");
        }
        catch(NoSuchAlgorithmException nsaex)
        {
            System.err.println(MD5Util.class.getName()
                + " 初始化失败,MessageDigest 不支持
MD5Util.");
            nsaex.printStackTrace();
        }
    }
    private static void appendHexPair (byte bt, StringBuffer
stringbuffer)
    {
        char c0=hexDigits[(bt&0xf0)>>4];
        char c1=hexDigits[bt&0xf];
        stringbuffer.append(c0);
        stringbuffer.append(c1);
    }
}

```

```

}
private static String bufferToHex(byte bytes[])
{
    return bufferToHex(bytes,0,bytes.length);
}
private static String bufferToHex(byte bytes[], int m, int n)
{
    StringBuffer stringbuffer=new StringBuffer(2*n);
    int k=m+n;
    for(int l=m;l<k;l++) appendHexPair(bytes[l],stringbuffer);
    return stringbuffer.toString();
}
public static String getMD5String(String s)
{
    return getMD5String(s.getBytes());
}
public static String getMD5String(byte[] bytes)
{
    messagedigest.update(bytes);
    return bufferToHex(messagedigest.digest());
}
//生成文件 MD5 校验码
public static String getFileMD5String (File file) throws
IOException
{
    InputStream fis;
    fis=new FileInputStream(file);
    byte[] buffer=new byte[1024];
    int numRead=0;

    while((numRead=fis.read(buffer))>0)
        messagedigest.update(buffer,0,numRead);
    fis.close();
    return bufferToHex(messagedigest.digest());
}
}

```

## 4 结语

程序经过测试，完成了目录同步的功能，可以在实际环境中使用。希望本开发实例对从事 Java 语言开发的读者有所帮助。

## 参考文献

- [1] 朱福喜，尹为民，余振坤. Java 语言与面向对象程序设计. 武汉大学出版社，2002，11.  
(收稿日期：2012-12-11)





## 对酒店音响设备自动开关的实现

刘同法

**摘 要:** 简述了利用波形音频低级输入设备函数打开波形音频设备获取音频信号, 通过音频信号读取系统时间再加上 30 秒的延时时间向前步进, 而后启用固定定时器获取系统标准时间与延时时间进行比较, 从而达到实现自动开关音响电源的目的。

**关键词:** 音频信号; Windows 消息队列; 低级音频设备; 单片机控制; 上位机命令

节约能源是存于这个自然界永恒的主题。那么实现几百瓦上千瓦的吧台包厢音响设备实施自动开关, 对于节能来说也是一个很有意义的事。音响设备随着声音的到来而开启, 随着声音的消失而关闭。要实现这样的一个动作, 只需要抓住两个关键要素, 一是捕获音频信号出现的信号点, 二是音响电源开关的执行者。捕获音频信号可以通过计算机的低层设备来实现, 因为 Windows 的低层是由消息队列组成。其最理想的方法就是找到 Windows 音频消息, 捕捉到这个消息便可以产生消息动作; 而实现音响自动开关动作只需要一个带串行的 8 引脚 8 位单片机即可, 当捕捉到音频信号后, 发送命令到单片机便可控制音响设备。

### 1 软件设计

#### 1.1 捕获音频信号

实现本功能一个最重要的一步就是抓取音频信号, 在 Windows 的消息队列中, 音频信号的消息是“MM\_WIM\_DATA”其捕获信号代码如下:

代码中调用 Windows 音频信号消息获取函数 WaveInProc 获取系统音频信号 MM\_WIM\_DATA 消息:

```
void CALLBACK WaveInProc (HWAVEIN waveOut, UINT
uMsg, DWORD dwInstance,
                        DWORD dwParam1, DWORD
dwParam2)
{ if( uMsg == MM_WIM_DATA )//如果是音频消息
    AMForm->ProcessInput();//读出音频信号并处理
}
//-----
//下面是 ProcessInput();函数的实体。
Void _fastcall TAMForm::ProcessInput()
{
    if( isStart ) //当音频输入设备打开时 isStart=true 即为真
    {
        WAVEHDR* in=&WaveHeaders[buff];
//音频文件头结构
```

```
//清除为输入音频信号准备的 WAVE 格式缓冲区
::wav elnUnprepareHeader ( hWaveIn, in, sizeof
(WAVEHDR) );
...
peakthread->Process(Peak, (short*)(in->lpData));
//启动音频峰值线程
buff++;
if( buff == buffers )
    buff=0;
SendHeader->dwBufferLength=WaveBufSize;
SendHeader->dwFlags=0;
SendHeader->dwLoops=0;
//为音频信号输入准备 WAVE 格式缓冲区
::waveInPrepareHeader ( hWaveIn, SendHeader,
sizeof(WAVEHDR) );
//为音频设备提供输入缓冲区
::waveInAddBuffer ( hWaveIn, SendHeader, sizeof
(WAVEHDR));
if( SendHeader == &WaveHeaders[buffers-1] )
    SendHeader = &WaveHeaders[0];
else
    SendHeader++;
}
}
//-----
//下面函数用于打开捕获音频信号设备
void _fastcall TAMForm::FormCreate(Tobject *Sender)
{ buffers = 2;
  buffer_size = 1024;
  buff = 0;
  //填充数据结构
  waveFormat.wFormatTag = WAVE_FORMAT_PCM;
//音频格式类型
  waveFormat.nChannels = 2; //通道个数(此处左右通道)
  waveFormat.nSamplesPerSec = 44100; //采样频率(此
//处为 44.1 千赫)
  waveFormat.wBitsPerSample = 16; //平均数据转移率
```





## PROGRAM LANGUAGE

```
//为 16 字节/s
waveFormat.nBlockAlign =
    (short) (waveFormat.wBitsPerSample/
8*waveFormat.nChannels);
waveFormat.nAvgBytesPerSec =
    waveFormat.nBlockAlign*waveFormat.
nSamplesPerSec;
waveFormat.cbSize = 0;
//启用四个线程
...
peakthread = new PeakThread( true, buffer_size );//创建
//捕获音频峰值信号线程

WaveHeaders = new WAVEHDR[buffers];
WaveBufSize = buffer_size*waveFormat.
wBitsPerSample/8*waveFormat.nChannels;
for( int i = 0; i<buffers; i++ )
{
    WaveHeaders[i].dwBufferLength = WaveBufSize;
    WaveHeaders[i].lpData = (char *)VirtualAlloc ( 0,
WaveBufSize, MEM_COMMIT,
                                PAGE_READWRITE );
}
//打开音频输入设备
if ( waveInOpen ( &hWaveIn, WAVE_MAPPER,
&waveFormat, (DWORD)WaveInProc,
                (DWORD)this, CALLBACK_FUNCTION ) ==
MMSYSERR_NOERROR)
{
    ...
    if ( waveInStart ( hWaveIn ) ==
MMSYSERR_NOERROR )
    {
        SendHeader = &WaveHeaders[buffers/2];
        buff = 0;
        isStart = true;
    } else {
        Application->MessageBox("错误信息", "多媒体错
误!", MB_OK);
        Application->Terminate();
    }
    ...
}
//-----
在音频信号峰值捕获线程函数中插入控制代码:
void __fastcall PeakThread::FillBuffer()
{
    .....
    view->Canvas->FillRect( rr );
    view->Canvas->Unlock();
    //当峰值达到 7000 以上时就发出探测到系统音频信号信息
    if(left>7000){ ChabanCtrlf->GetTime2(); }
    ...
    Suspend();
}
```

//-----

### 1.2 通过串行通信发送命令

该部分的工作原理是利用音频信号捕获的时刻启动定时器发出开启音响电源命令, 此后立即在定时器的运行函数中使用控制量锁定命令再一次发送, 样式如下:

```
if(m_bOpenYing) //由音频信号给予赋真值
{
    m_bOpenYing=false; //锁定命令的再一次发送
    Openchaban2(); //发送启用音响电源命令
}
```

关闭音响电源的办法是利用两个时间的时间差, A 时间只有时间原值由启动后的定时器读取, B 时间由时间原值加 30 秒延时后由捕获音频信号的函数调用, 此时如果系统中有音频信号出现, AB 时间开始产生并持续工作, 因为 B 时间在 A 时间的基础上加了 30 秒钟, 所以 A 时间总是比 B 时间小 30 秒。但是, 当系统的音频信号停止时 B 时间就静止下来了 (不能及时获得新的时间), 这时由定时器读取的 A 时间还在向前刷新, 当音频信号停下来足有 30 秒后, A 时间就会超过 B 时间, 当达到超过 31 秒时便立即下达关闭音响电源命令。这中间那怕早 1 秒也不会关闭电源。如果有必要可以将 30 秒时间再延长一点:

```
//下面就是读出时间原值函数(由定时器调用)
long int __fastcall TChabanCtrlf::GetTime() //读出系统时钟
{
    struct time m_time; //申请一个时间结构
    long nlscv=0; //用于存放秒时间
    gettimeofday(&m_time); //读出系统时间并存储于时间结构中
    ht.s=m_time.ti_hour; //时钟
    ht.f=m_time.ti_min; //分钟
    ht.m=m_time.ti_sec; //秒钟
    nlscv=ht.s*60*60+ht.f*60+ht.m; //化为秒用于方便比较
    return nlscv; //返回读出的秒时间(时间原值)
}
```

```
//-----
//下面就是读出的并在时间原值上加 30 秒钟函数
long int __fastcall TChabanCtrlf::GetTime1() //时间原值由
//GetTime2()读取
//计算比较时间并加上 30 秒延时时间(超前时间)
long nlscv=ht2.s*60*60+ht2.f*60+ht2.m+30;
return nlscv;
}
```

```
//-----
//下面也是读出时间原值函数(由音频信号捕获函数调用)
void __fastcall TChabanCtrlf::GetTime2()
{
    struct time m_time;
    gettimeofday(&m_time);
    ht2.s=m_time.ti_hour;
    ht2.f=m_time.ti_min;
    ht2.m=m_time.ti_sec;
    if(Timer1->Enabled==false) //如果定时器是关闭的
    {
        m_bOpenYing=true; //解定时器函数中的用于锁定命令发
        //送控制量(设为真)
    }
}
```





```

Timer1->Enabled=true; } //发现系统中有音频信号启用定
//时器进入工作状态
}
//-----
//下面是定时器函数体
void _fastcall TChabanCtrlf::Timer1Timer(TObject *Sender)
{
    if(m_bOpenYing)//如果捕获到音频信号就开锁
    { m_bOpenYing=false; //闭锁防止下次扫描再发命令
      OpenComm(); //打开串口发送开启电源命令

      if(GetTime()>GetTime1())//比较两个时间判断系统音频信号
      //是否停止
      //如果音频信号停止就关闭定时器
      if(Timer1->Enabled)Timer1->Enabled=false;
      CloseComm(); //发送关闭电源命令
      Label1->Caption="音乐已播放完全毕! ";
    }
}
//-----
//下面是发送打开电源命令函数
void _fastcall TChabanCtrlf::OpenComm()
{ Aing:
  if(cSerial.Open(cban.ttime.portNr,9600))
  { char c=0x7F; //0x7F 为打开电源命令
    cSerial.SendData(&c,1); //发送命令
    cSerial.Close(); //关闭串口
    Label1->Caption="打开电源命令已经发送";
  } else
  { m_nNum++;
    if(m_nNum>=200)goto Bing; else goto Aing;}
    //SendMessage("Not! COM1 没能打开! ");
Bing:
}
//-----
//下面是发送关闭电源命令函数
void _fastcall TChabanCtrlf::CloseComm()
{ Aing:
  if(cSerial.Open(cban.ttime.portNr,9600))
  { char c=0x7E; //0x7E 为关闭电源命令
    cSerial.SendData(&c,1); //发送命令
    cSerial.Close(); //关闭串口
    Label1->Caption="关闭电源命令已经发送";
  }
  else {m_nNum++; if(m_nNum>=200)goto Bing; else goto
Aing;}
  Bing:
}
//-----

```

## 1.3 应用程序界面

应用程序界面如图 1 所示。

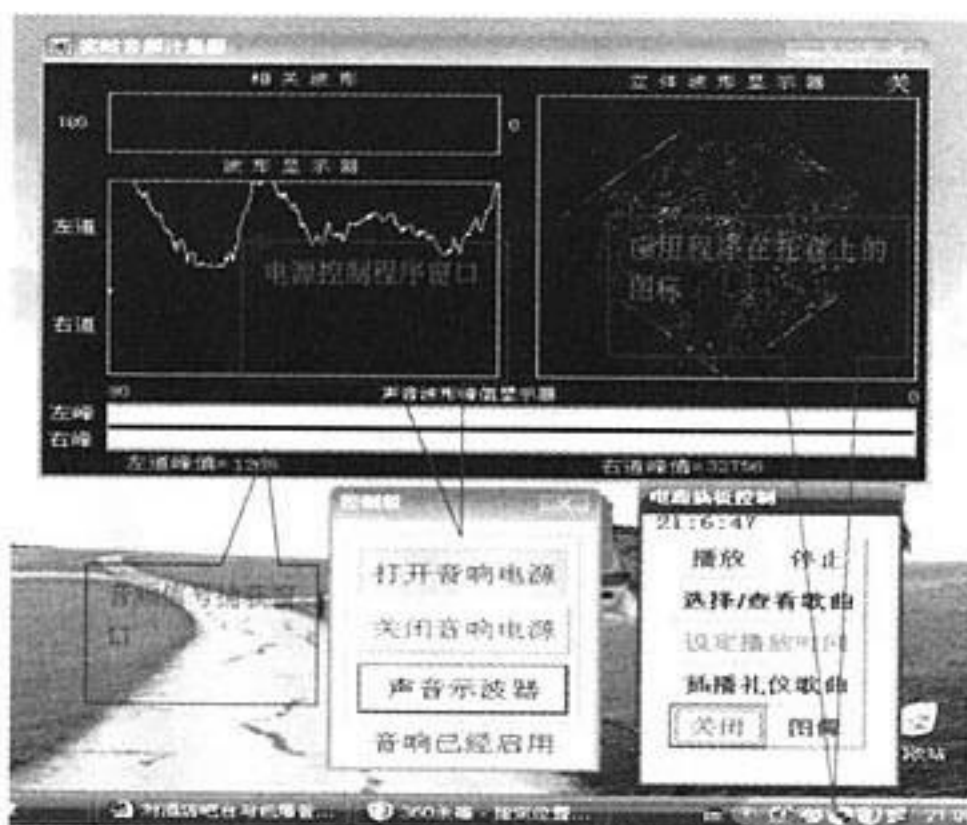


图 1 应用程序主要窗口界面

## 2 单片机控制系统

### 2.1 硬件

由单片机控制的电路如图 2 所示。

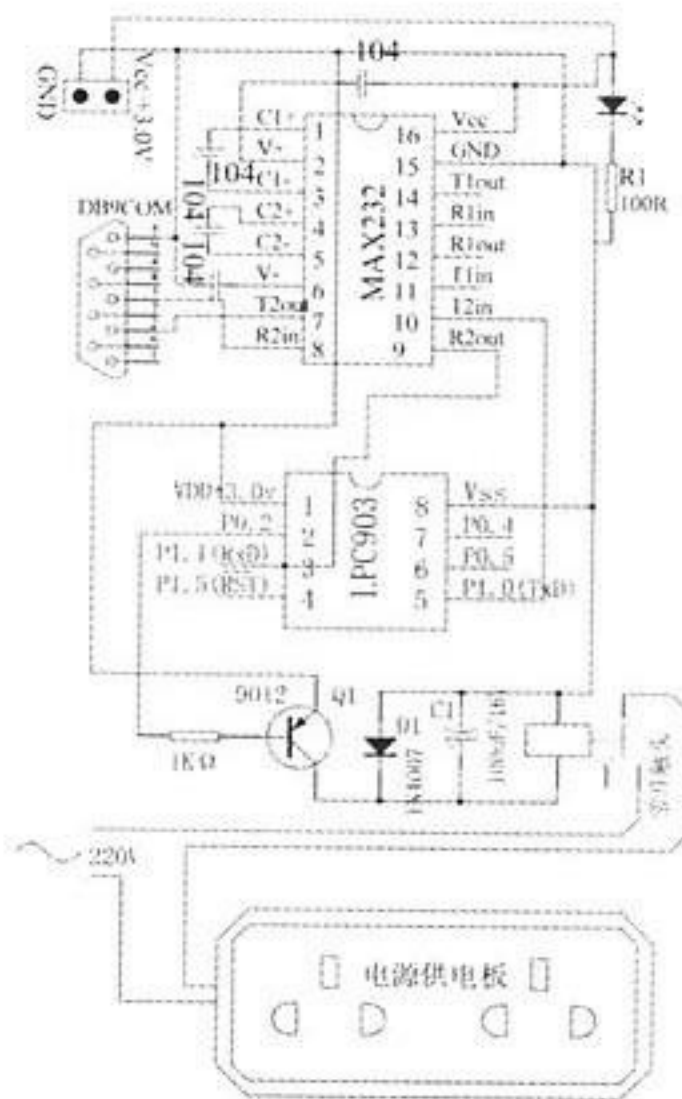


图 2 单片机控制电路图

### 2.2 软件

单片机使用的主程序如下：

```

//-----
...
sbit P02=P0^2; //用做控制电源开关
sbit P04=P0^4; //用做指示灯
(下转第 46 页)

```



# Android 平台 GPS 系统的应用开发

庞国明 隋瑞红

**摘要:** 根据 Android 平台的特点及 GPS 应用的性能需求, 开发基于 Android 平台、GPS 以及智能手机等技术为核心的 GPS 导航系统终端软件, 对系统平台和各主要功能模块设计进行了详细的研究和分析。

**关键词:** Android 系统; GPS 导航; 全球定位; 导航

## 1 引言

Android 作为 Google 移动互联网战略的重要组成部分, 将进一步推进“随时随地为每个人提供信息”这一企业目标的实现。Google 的目标是让移动通信不依赖于设备, 甚至是平台。出于这个目的, Android 将完善而不是替代 Google 长期以来推行的移动发展战略: 通过与全球各地的手机制造商和移动运营商成为合作伙伴, 开发既实用又有吸引力的移动服务, 并推广这些产品。

随着城市化的进展和家用轿车的普及, 原本很遥远的全球卫星定位系统 (Global Position System, GPS) 的使用越来越多 (如图 1 所示), 但车载导航仪高昂的价格和每次地图数据库升级所需的费用, 导致车载 GPS 导航产品仍旧没有非常普及。而且适用于在个人徒步行走的 GPS 实时导航产品还非常少。



图 1 GPS 导航应用

GPS 导航系统是集中应用了自动定位技术、地理信息系统 (GIS) 与数据库技术、计算机技术、无线通信技术的高科技综合系统。依据车载导航系统把车辆和道路综合起来考虑, 运用各种高新技术系统可以有效解决交通问题。目前随着智能手机的普及, 如何在智能手机中开发 GPS 导航系统可以说是目前的一个热点问题。

## 2 体系结构

### 2.1 Android SDK 架构

Android 平台由操作系统、中间件、用户界面和应用软件组成 (如图 2 所示)。它采用软件堆层的架构, 主要分为 3 部分。底层以 Linux 内核工作为基础, 由 C 语言开发, 只提供基本功

能; 中间层包括函数库 Library 和虚拟机 Virtual Machine, 由 C++ 开发。最上层是各种应用软件, 包括通话程序, 短信程序等, 应用软件则由各公司自行开发, 以 Java 作为编写程序的一部分。



图 2 Android 平台结构

第一层, 应用程序层, 该层提供一些核心应用程序包, 例如电子邮件、短信、日历、地图、浏览器和联系人管理等。同时, 开发者可以利用 Java 语言设计和编写属于自己的应用程序, 而这些程序与那些核心应用程序彼此平等、友好共处。

第二层, 应用程序框架层, 该层是 Android 应用开发的基础, 开发人员大部分情况是在和它打交道。应用程序框架层包括活动管理器、窗口管理器、内容提供者、视图系统、包管理器、电话管理器、资源管理器、位置管理器、通知管理器和 XMPP 服务 10 个部分。

第三层, 系统库和 Android 运行时, 系统库包括 9 个子系统, 分别是图层管理、媒体库、SQLite、OpenGLState、FreeType、WebKit、SGL、SSL 和 libc。Android 运行时包括核心库和 Dalvik 虚拟机, 前者既兼容了大多数 Java 语言所需要调用的功能函数, 又包括了 Android 的核心库, 比如 android.os、android.net、android.media 等等。后者是一种基于寄存器的 java 虚拟机, Dalvik 虚拟机主要是完成对生命周期的管理、堆栈的管理、线程的管理、安全和异常的管理以及垃圾回收等重要功能。

第四层, Linux 内核, Android 核心系统服务依赖于 Linux 2.6 内核, 如安全性、内存管理、进程管理、网络协议栈和驱动模型。Linux 内核也是作为硬件与软件栈的抽象层。包括显示驱动、摄像头驱动、键盘驱动、WiFi 驱动、Audio 驱动、





flash 内存驱动、Binder (IPC) 驱动、电源管理等。

## 2.2 开发环境

操作系统: Microsoft Windows 7

IDE: Eclipse Latefrom Version 3.3.1.1

开发包: Android SDK

模拟机: Android2.1 或 Android2.3 等

## 3 需求分析

### 3.1 GPS 导航应用系统规划设计

本设计的实现思路是在智能手机上开发一个 GPS 应用程序, 运行该程序后, 在其主界面上可以创建一个新的跟踪, 然后在后台启动一个 Service, 定时读取 GPS 数据获得用户目前所在的位置信息, 将其存入数据库中, 用户可以选择以往的跟踪记录, 将其轨迹显示在此, 由此可以记录用户去过的位置。

基于上面的思路, 基于 Android 的 GPS 导航应用开发的框图如图 3 所示。在智能手机的硬件平台上, 基于 android 手机操作系统来开发 GPS 导航应用。可以把这个 GPS 应用开发分为如下步骤: (1) 规划应用 UI。(2) 设计储存, 创建数据库、表。(3) 实现页面 Active 的跳转。(4) 实现服务 Service。(5) 完善和实现程序。(6) 程序打包、签名并发布。

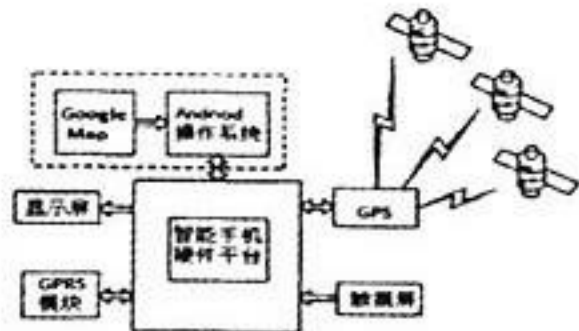


图 3 GPS 导航应用框架

### 3.2 性能

(1) 由于项目调用 android GPS api, 所以需要在 androidmanifest.xml 中写入开启调用 GPS API 权限的代码, 否则项目会因权限不够而无法运行调用 GPS API 的代码。

(2) 项目调用 GPS 后获取的地理位置手机, 受真实手机的 GPS 硬件系统、卫星状况、天气等指标的影响, 因此不同的手机运行本项目后效果不一样。

(3) 项目本身是 eclipse 环境下的 android 项目, 需要在已经添加了 android 插件的 eclipse 中以工程文件形式打开。

### 3.3 可靠性与可用性

本项目在 motorola defy+ 手机上测试可以正常运行, 因此在 android2.1 或更高版本的 android 真实手机上均可运行; 在模拟机方面, 系统版本为 android2.1 或者更高的版本并保证系统剩余内存为 1MB 的模拟机上都可正常运行。

## 4 系统实现

Android 在 GPS 导航系统软件设计过程中, 主要利用了

Android 软件平台的 5 大功能模块, 分别为 Android 应用程序层、应用框架层、普通函数程序库、Android 运行环境和 Linux 内核层。在应用程序层利用 Android 的各种组件 API 接口, 开发了针对 GPS 导航的 GPS、Google Map 电子地图的 Java 应用程序, 并由内核层提供了相应的底层驱动。

首先建立项目, 在 Eclipse 中依次选择 File→New Android Project 项, 输入项目名称和其他的辅助信息。下面具体进行 GPS 导航应用开发。

### 4.1 项目各界面实现

手机屏幕中的一个界面就是一个 Activity, 所以 Activity 可以算是 Android 当中最基本的模块之一, 一个 Android 应用程序可以认为就是由多个 Activity 的交互组成的, 这多个 Activity 之间可以进行相互跳转。不同的 Activity 实现不同的功能。

GPS 导航应用的界面主要有主界面, 地图定位界面, 新建跟踪界面, 个性化设置界面以及帮助界面等。主界面的目标是展示一个“以往跟踪的列表”, 下面以实现主界面为例进行设计。

```
<?xml version="1.0" encoding="utf-8"?>
android:orientation='vertical'
android:layout width="fill parent"
android:layout—height="fill — patent"
>
<TextView ??/>
<ListView??/>
</LinearLayout>
```

在 Android 中生成屏幕界面主要使用的是跨平台的扩展标记描述性语言 xml 进行配置生成的方式, 在 main.xml 使用了 LinearLayout 命令进行布局 (采用了垂直方式), 放置一个 TextView 和一个 ListView 组件, 其目的用来显示最近跟踪的列表。对于布局, 使用了“fill parent”, 因此在宽度和高度方向都是填充满父容器。

要让地图显示的话, 得将 MapView 加入到应用中来, 在布局文件 (main.xml) 中加入如下代码:

```
<com.google.android.maps.MapView
android:id="@+id/myGMap"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:enabled="true"
android:clickable="true"
android:apiKey="API_Key_String" />
```

### 4.2 GPS 定位功能实现

Android 操作系统支持 GPS API, 可以这样设置 Criteria 对象: Criteria c=new Criteria (); 创建一个 Criteria 对象, 接着将 Criteria 对象传递给 LocationManager 对象, LocationManager 对象提供了多种方式来获取经纬度坐标和其他数据, 它获取定位数据的方法如下: 对于经纬度, getLatitude () 返回纬度数据, getLongitude () 返回经度数据; 对于海拔高度, hasAltitude ()





## FORUM OF EXPERTS

判断是否有海拔数据，而 `getAltitude0` 则以米为单位返回海拔数据。

Android 支持地理定位服务的 API。该地理定位服务可以用来获取当前设备的地理位置。应用程序可以定时请求更新设备当前的地理定位信息。应用程序也可以借助一个 Intent 接收器来实现如下功能：以经纬度和半径划定的一个区域，当设备出入该区域时，可以发出提醒信息。

来看 `android.location` 包下一些比较重要的功能类，以下是几个 Android 关于定位功能的包中比较重要的类：

**LocationManager**：本类提供访问定位服务的功能，也提供获取最佳定位提供者的功能。另外，临近警报功能（前面所说的那种功能）也可以借助该类来实现。

**LocationProvider**：该类是定位提供者的抽象类。定位提供者具备周期性报告设备地理位置的功能。

**LocationListener**：提供定位信息发生改变时的回调功能。必须事先在定位管理器中注册监听器对象。

**Criteria**：该类使得应用能够通过向 `LocationProvider` 中设置的属性来选择合适的定位提供者。

Android 也提供了一组访问 Google MAP 的 API，借助 Google MAP 及定位 API，就能在地图上显示用户当前的地理位置：

Android 中定义了一个名为 `com.google.android.maps` 的包，其中包含了一系列用于在 Google Map 上显示、控制和层叠信息的功能类，以下是该包中最重要的几个类：

**MapActivity**：这个类是用于显示 Google MAP 的 Activity 类，它需要连接底层网络。

**MapView**：MapView 是用于显示地图的 View 组件。它必须和 MapActivity 配合使用。

**MapController**：MapController 用于控制地图的移动。

**Overlay**：这是一个可显示于地图之上的可绘制的对象。

**GeoPoint**：这个一个包含经纬度位置的对象。

下面来实际使用它们：

将来开发一个应用，它将能在 Google Map 上显示用户当前的位置。会通过使用 Google Map API 来显示地图，然后使用定位 API 来获取设备的当前定位信息以在 Google Map 上设置设备的当前位置。用户定位会随着用户的位置移动而发生改变。

**Activity**：

首先需要继承了一个继承了 MapActivity 的 Activity 类：

```
class MyGPSActivity extends MapActivity {
    ...
}
```

要成功引用 Google MAP API，必须先在 `AndroidManifest.xml` 中定义如下信息：

```
<uses-library android:name="com.google.android.maps" />
```

另外，要使用 Google Map 服务的话，需要一个 API key。

可以通过如下方式获取 API key：

(1) 找到 `USER_HOME\Local Settings\Application Data\Android` 目录下的 `debug.keystore` 文件。

(2) 使用 `keytool` 工具来生成认证信息 (MD5)，使用如下命令行：

```
keytool -list -alias androiddebugkey -keystore <
path_to_debug_keystore>.keystore -storepass
android -keypass android
```

(3) 打开 “Sign Up for the Android Maps API” 页面，输入之前生成的认证信息 (MD5) 后将获取到 API key。

(4) 替换上面 `AndroidManifest.xml` 配置文件中 “API\_Key\_String” 为刚才获取的 API key。

继续补全 `MyGPSActivity` 类的代码，以使用 `MapView`：

```
class MyGPSActivity extends MapActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        //创建并初始化地图
        gMapView = (MapView) findViewById(R.id.myGMap);
        GeoPoint p = new GeoPoint((int) (lat * 1000000), (int) (long
        * 1000000));
        gMapView.setSatellite(true);
        mc = gMapView.getController();
        mc.setCenter(p);
        mc.setZoom(14);
    }
}
```

另外，要使用定位信息的话，必须设置一些权限，`AndroidManifest.xml` 中的配置如下：

```
<uses-permission android:name="android.permission.
INTERNET"></uses-permission>
<uses-permission android:name="android.permission.
ACCESS_COARSE_LOCATION"></uses-permission>
<uses-permission android:name="android.permission.
ACCESS_FINE_LOCATION"></uses-permission>
```

使用定位管理器：

可以通过 `Context.getSystemService` 方法并传入 `Context.LOCATION_SERVICE` 参数获取定位管理器的实例：

```
LocationManager lm = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
```

之后，需要将原先的 `MyGPSActivity` 作一些修改，让它实现一个 `LocationListener` 接口，使其能够监听定位信息的变化：

```
class MyGPSActivity extends MapActivity implements
LocationListener {
    public void onLocationChanged(Location location) {}
    public void onProviderDisabled(String provider) {}
    public void onProviderEnabled(String provider) {}
    public void onStatusChanged (String provider, int status,
Bundle extras) {}
    protected boolean isRouteDisplayed() {}
}
```





```
return false;
}
}
```

添加一些代码，对 LocationManager 进行一些初始化工作，并在它的 onCreate () 方法中注册定位监听器：

```
@Override
public void onCreate(Bundle savedInstanceState) {
    LocationManager lm = (LocationManager)
        getSystemService (Context.LOCATION_SERVICE); lm.
        requestLocationUpdates (LocationManager.GPS_PROVIDER,
        1000L, 500.0f, this);
}
```

现在的代码中的 onLocationChanged 方法就会在用户的位置发生 500 米距离的改变之后进行调用。这里默认使用的 LocationProvider 是“gps” (GSP\_PROVIDER)，不过可以根据需要，使用特定的 Criteria 对象调用 LocationManger 类的 getBestProvider 方法获取其他的 LocationProvider。以下代码是 onLocationChanged 方法的参考实现：

```
public void onLocationChanged(Location location) {
    if (location != null) {
        double lat = location.getLatitude();
        double lng = location.getLongitude();
        p = new GeoPoint ((int) lat * 1000000, (int) lng *
        1000000);
        mc.animateTo(p);
    }
}
```

上面的代码用于获取当前的新位置并更新地图上的位置显示。还可以为应用程序添加一些诸如缩放效果、地图标注、文本等功能。

添加缩放控件：

// 将缩放控件添加到地图上

```
ZoomControls zoomControls = (ZoomControls)
gMapView.getZoomControls();
zoomControls.setLayoutParams (new ViewGroup.
LayoutParams(LayoutParams.WRAP_CONTENT,
LayoutParams.WRAP_CONTENT));
gMapView.addView(zoomControls);
gMapView.displayZoomControls(true);
```

添加 Map Overlay：

下面的代码是定义一个 overlay：

```
class MyLocationOverlay extends com.google.android.
maps.Overlay {
    @Override
    public boolean draw (Canvas canvas, MapView mapView,
    boolean shadow, long when) {
        super.draw(canvas, mapView, shadow);
        Paint paint = new Paint();
        // 将经纬度转换成实际屏幕坐标
```

```
Point myScreenCoords = new Point (); mapView.
getProjection().toPixels(p, myScreenCoords);
paint.setStrokeWidth(1);
paint.setARGB(255, 255, 255, 255);
paint.setStyle(Paint.Style.STROKE);
        Bitmap bmp = BitmapFactory.decodeResource
(getResources(), R.drawable.marker);
        canvas.drawBitmap (bmp, myScreenCoords.x,
myScreenCoords.y, paint);
        canvas.drawText ("Here I am ... ", myScreenCoords.x,
myScreenCoords.y, paint);
        return true;
    }
}
```

上面的这个 overlay 会在地图上显示一个“Here I am”的文本。把这个 overlay 添加到地图上去：

```
MyLocationOverlay myLocationOverlay = new
MyLocationOverlay();
List<Overlay> list = gMapView.getOverlays();
list.add(myLocationOverlay);
```

## 4.3 Google MAP 电子地图显示处理

利用 Android 平台开发导航地图过程中，主要采用 Android 提供的 MapView 和 MapActivity 这两个类实现。其中 MapView 是一个展示地图的视图，它可以获取键盘事件 (onKeyDown 和 onKeyUp) 来支持地图移动和缩放功能，同时也支持多层 Overlay，可在地图上画坐标、写地名、画图片等。MapView 只能通过 MapActivity 来建立，因为 MapView 需要在后台使用文件系统和网络，所有这些线程需要在 Activity 的生命周期中被控制。

## 5 结语

利用 android 操作系统在智能手机上实现了 GPS 导航的应用开发，利用终端各功能模块的整合，配合相应的软件驱动，完成指定的功能，如对定位信息进行数据处理，计算所在位置的经度、纬度、海拔、速度和时间等，并实时将定位的参数传输到服务器管理平台，方便管理人员对相关车辆进行实时监控与调度，提高 T 作效率和运营效益。因为 GPS 系统具有全天候、全球覆盖、三维定速、定时、高精度、快速、省时、高效率、应用广泛、多功能等特点，因此，可广泛应用于陆地、海洋以及航空航天等。而能实时更新 Google Map 地图的使用，更能直观地将丰富的城市地图、全国的公路网图、加油站、便利商店、政府机关、旅游景点、餐馆、医院、停车场等信息同步地在智能手机上显示出来，方便使用者 (不仅仅是汽车驾驶员) 进行查询参考，这是以后 Android 个性化应用开发，这将是一个潜力巨大的市场，会吸引无数软件开发厂商和开发者投身其中。

(收稿日期：2012-12-12)





# Oracle 程序包按自然方式自动维护分区

刘恒学

**摘 要:** 通过分析目前人工维护表分区的各种弊端, 提出了编写 Oracle 程序包按照自然方式实现自动维护分区的方法, 以达到减少人工维护量以及空闲空间占用量的目的。

**关键词:** Oracle 数据库; 范围分区; 间隔分区; PL/SQL 程序包

Oracle 10g 由于没有提供间隔分区的功能, 分区表的分区的增加需要人工定时进行。例如, 某些表按每月建一个范围分区, 那么在每年的某个月, 便会人工将下一年度的所有分区先创建出来, 以免到需要使用相应的分区时而相应的分区不存在而导致数据插入失败。每年需要人工修改生成下一年的脚本并执行, 容易发生错误, 而数据库管理员必须时刻惦记这件事, 这事变成了一块心病, 并且这种脚本一般考虑不会很周全, 当需要新增加分区表时, 每次需要修改脚本, 因此增加了数据库管理员的负担; 另外, 由于事先增加的表分区在很长时间不会存储数据, 故导致数据库的存储利用率不高。当然可以只建下一个月的分区以提高存储利用率, 但这样又进一步增加了数据库管理员的工作量, 每个月都需要进行人工增加分区的工作。Oracle 11g 提供了间隔分区的功能, 此时虽说分区可以自动维护, 但由于分区所在的表空间只能按循环方式选择在事先指定的多个表空间中的一个, 这样使得哪些分区具有相同的表空间不易控制, 不便于过时分区的迁移和清除工作。另外当分区表中没有即将插入的记录的分区, 数据库会增加相应的分区, 此时必然影响客户端的响应性能。分区的实质是减少管理数据的规模, 实际上应该存在一种根据空间尺寸来进行分区的间隔分区, 这是 Oracle 11g 所没有提供的, 可以将这种分区方式称之为空间间隔分区。这种分区的最后一个分区当数据量达到临界尺寸时, 便会自动增加一个新的分区以便接纳新的数据。因此如果能够通过编程让范围分区表按自然方式自动增加分区, 是一件值得进行的工作。

## 1 实现思路

先创建用于控制自动维护分区表的表分区和相关索引分区的配置表; 然后编写程序可以根据数据库中的现有分区表的目前状态生成各自的配置记录并插入到配置表中; 当然人工可以根据实际情况修改这些配置; 编写程序根据分区表的配置信息以及分区索引的配置信息, 增加分区表的分区, 通过控制分区表的表空间前缀可以做到不同分区表每个表分区在不同的表空间中, 也可以让它们的表空间前缀相同, 如果它们的时间间隔

相同的话, 可以将不同分区表的表分区放在相同的表空间中。通过利用 Oracle 的 DBMS\_SCHEDULER 包创建特定的任务定期执行按照自然的方式自动增加必要的分区的功能。

## 2 结构的设计

为了可以对自动增加分区进行灵活的人工控制, 创建一个用于配置分区表的相关参数的数据库基表 ALHXWS\_TABCONFIG, 其结构如下:

```
create table ALHXWS_TABCONFIG
(--分区表配置表
TABLE_OWNER VARCHAR2(30) not null, -- 分区表所属
--用户
TABLE_NAME VARCHAR2(30) not null, -- 分区表表名
P_TYPE NUMBER(2) not null, -- 范围分区类型::0 空间间
--隔 1 秒间隔 2 月间隔 3 数字间隔
INV_SUFFIX VARCHAR2(30), -- 固定后缀
TBS_PREFIX VARCHAR2(30), -- 表空间名前缀
PINTERVAL VARCHAR2(30), -- 分区间隔尺寸::根据范围
--分区类型具有不同的意义::
-- 0 空间间隔 不能根据目前情况计算出时所设定的秒间隔
-- 1 秒间隔 TO_DSINTERVAL
-- 2 月间隔 TO_YMINTERVAL
-- 3 数字间隔
INITSIZE NUMBER(14), -- 表分区初始尺寸
SUFFIX_FORMAT VARCHAR2(30), -- 后缀格式
SORTNO NUMBER, -- 排序序号
CONSTRAINT PK_ALHXWS_TABCONFIG PRIMARY
KEY (TABLE_OWNER, TABLE_NAME)
);
```

为了可以对分区表相关分区索引进行灵活的人工控制, 创建另一个用于配置分区表相关分区索引的相关参数的基表 ALHXWS\_INDCONFIG, 其结构如下:

```
create table ALHXWS_INDCONFIG
(--分区表分区索引配置表
INDEX_OWNER VARCHAR2(30) not null, -- 索引所属用户
INDEX_NAME VARCHAR2(30) not null, -- 分区表索引名
TBS_PREFIX VARCHAR2(30), -- 索引分区表空间
```





```
--名前缀
INITSIZE NUMBER(14), -- 索引分区初始尺寸
constraint PK_ALHXWS_INDCONFIG primary key
(INDEX_OWNER, INDEX_NAME)
);
```

另外，为了方便事后查询程序的执行情况，创建一个基表 ALHXWS\_LOG 用于记录程序执行时产生的日志，这个表本身为分区表，因此可以用来测试自动增加分区的功能，其结构如下：

```
create table ALHXWS_LOG
(--日志信息
LOGTIME TIMESTAMP (6) default systimestamp not null,
--产生时间
LOGTAG VARCHAR2(80), --分类标签
LOGINF VARCHAR2(4000) --日志内容
)
partition by range (LOGTIME)
(
partition P_2012 values less than (TIMESTAMP' 2013-01-01 00:00:00')
tablespace DATA_2012,
partition P_2013 values less than (TIMESTAMP' 2014-01-01 00:00:00')
tablespace DATA_2013
);
create index I_ALHXWS_LOG_LOGTIME on ALHXWS_LOG
(LOGTIME);
```

## 3 PL/SQL 程序包的分析与设计

### 3.1 合理假设

为了降低自动维护分区的复杂性以及简化程序包的编写，并具有足够宽广的适用范围，特作下列假设：

(1) 考虑只有一个 DATE 或 TIMESTAMP 或数字类型字段的范围分区。

(2) 表分区名与索引分区名名字相同，分区名的命名规则为分区前缀（统一为 P\_）+分区名后缀，并且分区名后缀为定长字符串。

(3) 考虑到 DATE 与 TIMESTAMP 的统一，分区的分界值统一为 TIMESTAMP'YYYY-MM-DD HH24:MI:SS.FF9'。因为 Oracle 会根据分区字段的类型自动将此格式转化为相应的格式。

(4) 表空间名为表空间前缀（每个表可以不同）+分区名后缀，假定此处的分区名后缀与分区名中的分区名后缀相同。

### 3.2 初始配置

按照数据库中当前的分区表的元数据初始化上述配置表。

### 3.3 表分区与表空间初始尺寸

一般在建数据库之前会根据经验进行所谓的数据库规划，事先确定表空间和表分区的初始尺寸。然而实践证明，实际的数据往往与事先的数据库规划有较大的出入，因此，实现一种

机制，能够根据近期的数据量来确定最近的新的数据分区的初始尺寸以及所在表空间的初始尺寸。

### 3.4 自动增加表分区

根据两个分区表及索引的配置信息，依据任务的执行时间间隔执行自动增加分区的功能。

### 3.5 分区的合并、移动、清除

为了方便测试以及过时的数据的迁移和清理，为了将已经生成的表分区与自动生成的表分区样式一致，以及今后人工的维护，实现合并分区、移动分区及清除分区的功能。

### 3.6 分区表与非分区表之间的转换

为了根据数据库的运行情况，调整表的存储方式，实现分区表和非分区表之间相互转换的功能。

### 3.7 异常情况的报警与处理

为了方便数据库管理员的检查，可以通过 dbms\_system.ksdwrt 将程序包执行时产生的重要信息输出到数据库日志文件 alert\_sid.log 文件上，以及在日志表上插入日志记录。

在系统出现异常时导致的执行失败，人工在部分成功的基础上再执行一次可以保证完全成功。

### 3.8 自动生成 3 个相关的表

当程序包需要使用基表 ALHXWS\_TABCONFIG、ALHXWS\_INDCONFIG 和 ALHXWS\_LOG 时，如果不存在，程序则自动创建这些表。

### 3.9 执行的控制

为了实现对自动任务的控制，设计了自动分区任务的启动、暂停、终止功能。

### 3.10 分区表插入异常的处理

另外也可以根据实际情况，在分区表当插入记录时发生出现 ORA-14400 (插入的分区关键字未映射到任何分区) 错误时调用程序包中的 step 函数增加相应的分区，然后再插入之前插入失败的记录。

## 4 程序包使用

### 4.1 在 SYS 用户下创建与执行程序包

在用户 SYS 下创建与执行 PL/SQL 程序包，避免在其他用户下由于权限不够需要特殊授权的麻烦。因为既然是对分区表的维护，自然具有 SYS 的口令与权限，故能够在 SYS 用户下操作。

### 4.2 多个分区表共用表空间

对于数据量较小的分区表，建议共用一个相同的表空间前缀，这样当它们的分区名相同时，其后缀相同，于是它们的数据会存放在相同的表空间中，以免产生过多的数据库文件。当然那些数据量特别大的分区表，建议使用独立的表空间前缀，比如表名为 MYGOD，那么其表分区的表空间前缀可以为





## -----DATABASE-----

DMYGOD\_，而其分区索引的表空间前缀可以为 IMYGOD\_。

### 4.3 估计可能产生数据库数据文件的最大个数

不要使数据库文件个数超过由数据库参数 DB\_FILES 所规定的值。以免不能创建新的表空间。为此在使用此程序包之前，需要根据分区配置情况计算可能产生数据库数据文件的最大个数，并注意设置 DB\_FILES 到合适的值。

### 4.4 使用步骤

(1) 在 SYS 用户下创建分区维护程序包。

(2) 初始化：执行程序包中的初始化配置表函数 initconfig，此时此函数如果发现没有程序包相关 ALHXWS\_TABCONFIG、ALHXWS\_INDCONFIG 和 ALHXWS\_LOG，程序会自动创建这 3 个表，然后根据数据库中的实际情况将各种满足一定条件的分区表及相应的分区索引插入到 ALHXWS\_TABCONFIG、ALHXWS\_INDCONFIG，并且根据分区表的相关情况配置这些表中记录的相应字段的值。

(3) 人工配置：查询出自动配置好的各个分区表及其分区索引相关的记录，人工根据实际情况给以必要的修改，以便分区表按照人工的设置来自动维护。

(4) 启动：在配置好各个分区表和分区索引的相关参数后，可以执行函数 runstep 来执行分区自动维护的任务。在执行此函数如果发现相应的 JOB 还未创建，它会自动创建它。

(5) 停止：在某些情况下如果需要停止 JOB 特定一段时间的运行，可以执行函数 disablestepjob。

(6) 查询 ALHXWS\_LOG 表看是否有异常的地方，如果发现本应增加的分区而未增加，可以人工执行 runstep 加以补救。

## 5 源程序

### 5.1 包头

```
CREATE OR REPLACE PACKAGE alhxws_steppartition
```

```
--jingaideliulao 2012-09-18
```

```
IS
```

```
-- Author : jingaideliulao
```

```
-- Created : 2012-09-18 09:18:09
```

```
-- Purpose : 分区维护
```

```
-- Public type declarations
```

```
TYPE TCRRecTyp IS RECORD (
```

```
    szUser VARCHAR2(30),
```

```
    szTABLE_NAME VARCHAR2(30),
```

```
    nP_TYPE NUMBER(2),
```

```
    szINV_SUFFIX VARCHAR2(30),
```

```
    szTBS_PREFIX VARCHAR2(30),
```

```
    szPINTERVAL VARCHAR2(30),
```

```
    nSIZE NUMBER(14),
```

```
    szSUFFIX_FORMAT VARCHAR2(30));
```

```
-- Public constant declarations
```

```
--<ConstantName> constant <Datatype> := <Value>;
```

```
-- Public variable declarations
```

```
--<VariableName> <Datatype>;
```

```
-- Public function and procedure declarations
```

```
--将分区表按照合适的情况增加表分区
```

```
PROCEDURE step ;
```

```
--将给定的分区表按照其配置增加表分区
```

```
FUNCTION step ( tcrec TCRRecTyp) RETURN INTEGER;
```

```
--将给定的分区表按照合适的情况增加表分区
```

```
FUNCTION step ( szUser VARCHAR2,tname VARCHAR2 )  
RETURN INTEGER;
```

```
--获取分区表如何合适增加表分区的规划--分区表表名--新
```

```
--增加的分区名的后缀--用于分裂的分区名的后缀
```

```
FUNCTION getaddpartitionplan ( tcrec IN OUT TCRRecTyp,  
p_name OUT VARCHAR2, tbs_name OUT VARCHAR2, attm  
OUT TIMESTAMP_UNCONSTRAINED, sp_name OUT VAR  
CHAR2,stbs_name OUT VARCHAR2) RETURN INTEGER;--  
0 增加表分区 1 分裂表分区 2 保持不变
```

```
--获取分区表如何合适增加表分区的规划--分区表表名--新
```

```
--增加的分区名的后缀--用于分裂的分区名的后缀
```

```
FUNCTION getaddpartitionplan_old ( tcrec IN OUT  
TCRRecTyp, p_name OUT VARCHAR2, tbs_name OUT  
VARCHAR2, attm OUT TIMESTAMP_UNCONSTRAINED,  
sp_name OUT VARCHAR2,stbs_name OUT VARCHAR2)  
RETURN INTEGER;--0 增加表分区 1 分裂表分区 2 保持不变
```

```
--获取固定尺寸分区表的下一个时间间隔
```

```
FUNCTION calculateinterval ( tcrec IN OUT TCRRecTyp,  
p_pos INTEGER,ilInterval OUT DSINTERVAL_UNCONSTRAI  
NED ) RETURN INTEGER;
```

```
--修剪时间间隔::没必要在小的数量上斤斤计较
```

```
FUNCTION pruninginterval ( tcrec IN OUT TCRRecTyp,  
ilInterval DSINTERVAL_UNCONSTRAINED ) RETURN  
DSINTERVAL_UNCONSTRAINED;
```

```
--创建表空间
```

```
FUNCTION createtbs ( tbs_name VARCHAR2, tbs_bytes  
NUMBER ) RETURN INTEGER;
```

```
--创建表空间::带尺寸单位
```

```
FUNCTION createtbs ( tbs_name VARCHAR2, tbs_bytes  
VARCHAR2 ) RETURN INTEGER;
```

```
--增加表分区
```

```
FUNCTION addstep ( tcrec TCRRecTyp, p_name  
VARCHAR2 ,tbs_name VARCHAR2 ,attm TIMESTAMP_UNC  
ONSTRAINED) RETURN INTEGER;
```

```
--分裂表分区
```

```
FUNCTION splitstep ( p_suffix VARCHAR2,sp_suffix  
VARCHAR2) RETURN INTEGER;
```

```
--分裂表分区
```

```
FUNCTION splitstep (tcrec TCRRecTyp,p_name VARCHAR2,  
tbs_name VARCHAR2, attm TIMESTAMP_UNCONSTRAINE  
D, sp_name VARCHAR2,stbs_name VARCHAR2 ) RETURN  
INTEGER;
```

```
--具体分裂表分区
```

```
FUNCTION splitpartition (szUser VARCHAR2,tname
```





```

VARCHAR2,p_name VARCHAR2,tbs_name VARCHAR2,
attm TIMESTAMP_UNCONSTRAINED, sp_name VARCHAR
2 ,stbs_name VARCHAR2) RETURN INTEGER;

```

--合并表分区

```

FUNCTION mergepartition ( p_name VARCHAR2,
mp_name VARCHAR2 ) RETURN INTEGER;

```

--合并表分区

```

FUNCTION mergepartition (szuser VARCHAR2,tname
VARCHAR2,p_name VARCHAR2,mp_name VARCHAR2 )
RETURN INTEGER;

```

--获取分区表的表空间的合适初始尺寸

```

FUNCTION gettbsoksize (tbs_prefix VARCHAR2,
suffix_format VARCHAR2,tbs_size OUT NUMBER ) RETURN
INTEGER;

```

--将表空间增加到表的表分区中

```

FUNCTION addtpartition (szUser VARCHAR2,tname
VARCHAR2, p_name VARCHAR2 ,tbs_name VARCHAR2,
attm TIMESTAMP_UNCONSTRAINED) RETURN INTEGER;

```

--重建索引分区

```

FUNCTION rebuildipok (szUser VARCHAR2, iname
VARCHAR2, p_suffix VARCHAR2 ) RETURN INTEGER;

```

--重建索引分区

```

FUNCTION rebuildip (szUser VARCHAR2, iname
VARCHAR2, p_suffix VARCHAR2 ,itbs_prefix VARCHAR2)
RETURN INTEGER;

```

--重建表相关的索引分区

```

FUNCTION rebuildip (szUser VARCHAR2,tname
VARCHAR2 ) RETURN INTEGER;

```

--重建所有索引分区

```

FUNCTION rebuildip RETURN INTEGER;

```

--重建所有索引分区

```

FUNCTION rebuildipall RETURN INTEGER;

```

--检查表分区是否有记录

```

FUNCTION checkrow_p ( szUser VARCHAR2,tname in
varchar2, p_name in varchar2 ) return boolean;

```

--快速清除表分区的内容

```

FUNCTION trunc_p ( szUser VARCHAR2,tname in varchar2,
p_name in varchar2 ) return integer;

```

--移动表分区

```

FUNCTION move_tp ( p_name in varchar2 ) return integer;

```

--移动表分区

```

FUNCTION move_tp ( tcrec TCRRecTyp, p_name in varchar2
,old_tbs_name varchar2) return integer;

```

--表分区是否存在

```

FUNCTION canmovetp (tcrec TCRRecTyp, p_name in
varchar2,old_tbs_name out varchar2) return boolean;

```

--删除表分区

```

FUNCTION drop_tp ( p_name in varchar2 ) return integer;

```

--删除表分区

```

FUNCTION drop_tp ( tcrec TCRRecTyp, p_name in varchar2 )
return integer;

```

--删除表分区

```

FUNCTION drop_ip ( szUser VARCHAR2,iname in varchar2,
p_name in varchar2 ) return integer;

```

--删除表空间

```

FUNCTION drop_tbs ( tbs in varchar2) return integer;

```

--删除表空间

```

FUNCTION drop_tbs ( tbs in varchar2,tbs_prefix in
varchar2) return integer;

```

--删除已经删除表分区的索引表空间

```

FUNCTION drop_itbs (szUser VARCHAR2,tname in
varchar2, p_name in varchar2) return integer;

```

--删除无用的表空间

```

FUNCTION drop_tbses return integer;

```

--获取表空间文件名

```

FUNCTION getfilename ( tbs_name in varchar2 ) return
VARCHAR2;

```

--初始化配置表

```

FUNCTION initconfig return integer;

```

--初始化所有分区表配置表

```

FUNCTION inittabconfig return integer;

```

--初始化特定分区表配置表

```

FUNCTION inittabconfig (szUser VARCHAR2, tname
VARCHAR2 ) return integer;

```

--初始化所有分区索引配置表

```

FUNCTION initindconfig return integer;

```

--初始化特定分区表分区索引配置表

```

FUNCTION initindconfig (szUser VARCHAR2, tname
VARCHAR2 ) return integer;

```

--将 Date 表达式转换为 timestamp 表达式

```

FUNCTION totimestamp ( dt varchar2 ) return varchar2;

```

--求时间戳表达式的值

```

FUNCTION mytimestamp (szexp varchar2) return
TIMESTAMP_UNCONSTRAINED;

```

--比较两个时间戳找出之间的关系::0 无 1 秒的小数部分相同  
2 秒相同 3 分相同 4 小时相同 5 天相同 6 月相同

```

FUNCTION cmptimestamp (t1
TIMESTAMP_UNCONSTRAINED,t2
TIMESTAMP_UNCONSTRAINED) return integer;

```

--取固定后缀

```

FUNCTION inv_suffix(t TIMESTAMP_UNCONSTRAINED,ist
integer) return varchar2;

```

--取字符串出现第一个数字字符的前缀

```

FUNCTION inv_prefix(sz varchar2) return varchar2;

```

--取两个字符串开头相同部分

```

FUNCTION inv_prefix (sz1 varchar2,sz2 varchar2) return
varchar2;

```

--取规定长度数值的字符串

```

FUNCTION my_to_char (i integer,len integer) return
varchar2;

```

--获取秒数

```

FUNCTION getsecond ( ids DSINTERVAL_
UNCONSTRAINED) return decimal;

```

--创建日志表





## .....DATABASE.....

```

FUNCTION createtab_alhows_log return integer;
--创建分区表配置表
FUNCTION createtab_alhows_tabconfig return integer;
--创建索引分区表配置表
FUNCTION createtab_alhows_indconfig return integer;
--立即执行增加分区任务
PROCEDURE runstep;
--创建执行增加分区任务
PROCEDURE createstepjob;
--删除增加分区任务
PROCEDURE dropstepjob;
--停止增加分区任务
PROCEDURE stopstepjob;
--以后不再运行增加分区任务
PROCEDURE disablestepjob;
--以后可以定时运行增加分区任务
PROCEDURE enablestepjob;
--写日志
PROCEDURE writelog(sztag VARCHAR2,szlog VARCHAR2 );
--写日志
PROCEDURE writelog ( ilevel integer, sztag VARCHAR2,
szlog VARCHAR2 );
END alhows_steppartition;

```

### 5.2 包体

由于包体过长，此处只列出其中两个函数，其他函数与过程略。

```

--将给定的分区表增加表分区使其能够插入给定的记录
FUNCTION step
(
szUser VARCHAR2, --表的用户名
tname VARCHAR2, --表名
insertt TIMESTAMP_UNCONSTRAINED --所插入的记录的
--日期时间值
)
RETURN INTEGER
is
iResult integer;
tcrec TCRRecTyp;
--新分区名
v_p_name VARCHAR2(30);
v_p_name_old VARCHAR2(30);
--新分区表空间名
v_tbs_name VARCHAR2(30);
v_attm TIMESTAMP_UNCONSTRAINED;
v_attm_old TIMESTAMP_UNCONSTRAINED;
--分裂分区名
v_sp_name VARCHAR2(30);
--分裂分区表空间名
v_stbs_name VARCHAR2(30);
begin
iResult:=0;

```

```

execute immediate 'SELECT TABLE_OWNER,
TABLE_NAME, P_TYPE, INV_SUFFIX, TBS_PREFIX,
PINTERVAL,INITSIZE,SUFFIX_FORMAT FROM ' || szTabUser
|| '.ALHWS_TABCONFIG WHERE TABLE_OWNER=:1 AND
TABLE_NAME=:2'
into tcrec USING szuser,tname;
v_p_name_old:=v_p_name;
v_attm_old:=TIMESTAMP'0001 -01 -01 00:00:
00.0000000000';
LOOP
iResult := getaddpartitionplan (tcrec, v_p_name,
v_tbs_name, v_attm, v_sp_name,v_stbs_name);
if v_p_name_old=v_p_name or v_attm_old=v_attm or
v_p_name is null or v_tbs_name is null then --避免不作为时
--无限循环
EXIT;
end if;
CASE iResult
WHEN 0 THEN
iResult:=addstep(tcrec,v_p_name,v_tbs_name,v_attm);
if 0!=iResult then
EXIT;
end if;
ELSE EXIT;
END CASE;
v_p_name_old:=v_p_name;
v_attm_old:=v_attm;
if v_attm>insertt then
EXIT;
end if;
END LOOP ;
return (iResult);
EXCEPTION
WHEN OTHERS
THEN
iResult:=SQLCODE;
writelog('step4', SUBSTR (SQLERRM, 1, 300));
return (iResult);
end step;
--获取分区表如何合适增加表分区的规划
FUNCTION getaddpartitionplan
(
tcrec IN OUT TCRRecTyp, --分区表配置信息
p_name OUT VARCHAR2, --新分区名
tbs_name OUT VARCHAR2, --新分区表空间名
attm OUT TIMESTAMP_UNCONSTRAINED,--新分区
--分界点
sp_name OUT VARCHAR2, --分裂分区名
stbs_name OUT VARCHAR2 --分裂分区表空间名
)
RETURN INTEGER --0 增加表分区 1 分裂表分区 2 保持

```





```
--不变
is
iResult integer;
iMaxpos integer;
iDays integer;
v_hv VARCHAR2(255);
v_partitions_name VARCHAR2(80);
v_partitions_name_next VARCHAR2(80);
v_high_value TIMESTAMP_UNCONSTRAINED;
v_high_value_next TIMESTAMP_UNCONSTRAINED;
v_old TIMESTAMP_UNCONSTRAINED;
v_new TIMESTAMP_UNCONSTRAINED;
v_next TIMESTAMP_UNCONSTRAINED;
iInterval DSINTERVAL_UNCONSTRAINED;
CURSOR c_partitions( v_tname varchar) IS
select partition_name,tablespace_name,high_value,PARTITION_POSITION
from dba_tab_partitions
where table_name=v_tname and table_owner=tcrc.szUser
ORDER BY PARTITION_POSITION DESC;
begin
iResult:=0;
--求当前表分区最大序号
select max(PARTITION_POSITION)
into iMaxpos
from dba_tab_partitions
where table_name =tcrc.szTABLE_NAME and
table_owner=tcrc.szUser ;
--根据当前表分区最大序号查询
select partition_name,tablespace_name,high_value
into sp_name,stbs_name,v_hv
from dba_tab_partitions
where table_name=tcrc.szTABLE_NAME and table_owner=
tcrc.szUser and PARTITION_POSITION=iMaxpos;
if UPPER(v_hv)='MAXVALUE' then
v_high_value_next:=TIMESTAMP'9999 -12 -30 23:59:
59.999999999';
else
v_high_value_next:=mytimestamp(v_hv);
end if;
--如果为 0 空间间隔则求出相应间隔
if 0=tcrc.nP_TYPE then
iResult:=calculateinterval(tcrc,iMaxpos,iInterval);
iInterval:=pruninginterval(tcrc,iInterval);
end if;
if iMaxpos>1 then
select high_value
into v_hv
from dba_tab_partitions
where table_name =tcrc.szTABLE_NAME and
table_owner =tcrc.szUser and PARTITION_POSITION =
```

```
iMaxpos -1;
v_high_value:=mytimestamp(v_hv);
else
v_high_value:=systimestamp;
end if;
v_hv:=to_char (v_high_value,'YYYY-MM-DD HH24:MI:SS.
FF9');
v_old:=TO_TIMESTAMP(substr(v_hv,1,length(v_hv) -length
(tcrc.szINV_SUFFIX))
|| tcrc.szINV_SUFFIX,'YYYY-MM-DD HH24:MI:SS.FF9');
CASE tcrc.nP_TYPE
WHEN 1 THEN --1 秒间隔
BEGIN
v_new:=v_old+TO_DSINTERVAL(tcrc.szPINTERVAL);
v_next:=v_high_value +TO_DSINTERVAL (tcrc.
szPINTERVAL);
END;
WHEN 2 THEN --2 月间隔
BEGIN
v_new:=v_old+TO_YMINTERVAL(tcrc.szPINTERVAL);
v_next:=v_high_value +TO_YMINTERVAL (tcrc.
szPINTERVAL);
END;
ELSE --0 空间间隔
BEGIN
v_new:=v_old+iInterval;
v_next:=v_high_value+iInterval;
END;
END CASE;
if v_new<v_high_value_next and (iMaxpos>1 and v_new >
v_high_value or iMaxpos<=1) and v_next<v_high_value_next
then
iResult := 1;
p_name:=szPT_prefix || to_char (v_old,tcrc.
szSUFFIX_FORMAT);
tbs_name:=tcrc.szTBS_PREFIX || to_char (v_old, tcrc.
szSUFFIX_FORMAT );
atm:=v_new;
end if ;
--判断是否保持不变
if iResult=1 then
select extract (day from v_new - systimestamp) into
iDays from dual;
else
select extract (day from v_high_value_next -
systimestamp) into iDays from dual;
end if;
if iDays>iDayInterval then
iResult:=2;--保持不变
end if;
if iResult = 0 then
```





## DATABASE

```

v_hv:=to_char(v_high_value_next,'YYYY-MM-DD HH24:
MI:SS.FF9');
if 0=tcrec.nP_TYPE then
    v_old:=v_high_value_next;
else
    v_old:=TO_TIMESTAMP(substr(v_hv,1,length(v_hv) -
length(tcrc.szINV_SUFFIX),'YYYY-MM-DD HH24:MI:SS.FF9');
end if;
CASE tcrc.nP_TYPE
    WHEN 1 THEN --1 秒间隔
        BEGIN
v_new:=v_old+TO_DSINTERVAL(tcrc.szPINTERVAL);
        END;
    WHEN 2 THEN --2 月间隔
        BEGIN
v_new:=v_old+TO_YMINTERVAL(tcrc.szPINTERVAL);
        END;
    ELSE --0 空间间隔
        BEGIN
            v_new:=v_old+ilInterval;
        END;
END CASE;
p_name:=szPT_prefix || to_char(v_old,tcrc.
szSUFFIX_FORMAT);
tbs_name:=tcrc.szTBS_PREFIX || to_char(v_old,tcrc.
szSUFFIX_FORMAT);
attn:=v_new;
end if;
if 1=iResult and p_name=sp_name then
    p_name:=p_name || '0';
    tbs_name:=tbs_name || '0';
end if;
return iResult;
EXCEPTION
    WHEN OTHERS
    THEN
        iResult:=SQLCODE;
        writelog('getaddpartitionplan', SUBSTR (SQLERRM, 1,
300));
        return (iResult);
end getaddpartitionplan;
--获取固定尺寸分区表的下一个时间间隔
FUNCTION calculateinterval
(
    tcrc IN OUT TCRcTyp,
    p_pos INTEGER,
    ilInterval OUT DSINTERVAL_UNCONSTRAINED
)
RETURN INTEGER
is

```

```

iResult integer;
iSize NUMBER(11);
iPSize NUMBER(11);
p_name VARCHAR2(30);
t DECIMAL(20,9);
t1 DECIMAL(20,9);
i integer;
v_high_value_old TIMESTAMP_UNCONSTRAINED;
v_high_value_new TIMESTAMP_UNCONSTRAINED;
sql_stmt VARCHAR2(300);
CURSOR c_partitions( v_pos integer) IS
select partition_name,high_value,tablespace_name
from dba_tab_partitions
where table_name =tcrc.szTABLE_NAME and
table_owner=tcrc.szUser and partition_position<=v_pos
ORDER BY PARTITION_POSITION DESC;
begin
    --t = t1*iSize/iPSize
    iResult:=0;
    ilInterval:=TO_DSINTERVAL(tcrc.szPINTERVAL);
    i:=0;
    FOR v_partitions IN c_partitions(p_pos) LOOP
        if checkrow_p (tcrc.szUser,tcrc.szTABLE_NAME,
v_partitions.partition_name) then
            v_high_value_new:=mytimestamp ( totimestamp (
v_partitions.high_value));
            if 1=i then
                p_name:=v_partitions.partition_name;
            end if;
            if 2=i then
                t1:=getsecond(v_high_value_old -v_high_value_new) ;
                exit;
            end if;
            v_high_value_old:=v_high_value_new;
            i:=i + 1;
        else
            i:=0;
            t1:=null;
            v_high_value_old:=null;
        end if;
    END LOOP;
    if 2=i and t1 is not null then
        sql_stmt:='select sum(bytes)
from dba_segments
where owner = :1 and segment_name = :2 and
partition_name=:3
group by owner,segment_type,tablespace_name,
segment_name,partition_name ' ;
        execute immediate sql_stmt into iPSize USING tcrc.
szUser,tcrc.szTABLE_NAME,p_name;
(下转第 54 页)

```



# 用 Excel VBA 灵活复制照片并修改文件名

张貽忠 洪冰

**摘要:** 个人信息中经常需要照片, 而采集照片都要有文件名, 可以使用顺序号、顺序号 + 姓名、身份证号码等 (因姓名有可能是重名故不能单纯使用姓名)。为了加快采集照片的速度, 一般使用顺序号。采集后, 为确保人和顺序号照片的一致性, 需要对照片进行修改文件名核对, 重复性的工作量大, 用 Excel VBA 编写一个程序来复制照片并修改文件名, 不仅快而且准确。

**关键词:** Excel VBA 编程; 复制照片; 修改文件名

## 1 设计思路

(1) 照片采集时不管是相机软件还是计算机软件, 都会自动形成文件名 (文件类型为 JPG 格式)。采集照片后, 先制作一个照片顺序号与姓名、身份证号码等其他信息 (用于复制照片后修改照片文件名) 的一一对应的数据表, 然后导入系统中的“数据”工作表中, 导入后程序自动对数据设置最适合的列宽、添加框线、冻结窗口等。

(2) 如果有身份证号码信息, 可自动查找“身份证号码”列, 并验证身份证号码的正确性, 因为身份证号码终身不变, 所以可以用“身份证号码”作为照片的永久文件名, 将照片复制到“身份证号码照片”文件夹下, 因为身份证号码不会重复, 所以此文件夹下不必设置照片分类子文件夹, 并永久保存。而“原照片”文件夹可以保持不动, 作为原始数据。如果发现缺照片, 能在数据清单后面的空列中自动添加“缺照片”信息。

(3) 如果没有身份证号码信息, 可以按需要复制照片并按需要修改文件名。

(4) 采用隐藏“常规菜单”, 显示“自定义菜单”格式操作程序 (如图 1 所示)。

导入信息	验证身份证号	处理照片名	关闭系统
导入信息	验证号码的正确性	生成以身份证号为文件名的照片	关闭系统
		从以身份证号为文件名的照片中提取照片	
		按需要提取照片	

图 1 自定义菜单

(5) 为了便于核对照片与本人及信息是否相符, 可以将照片名替换成“序号 + 姓名 + 身份证号”, 然后在资源管理器窗口的“缩略图”查看方式下使用截图软件截图, 再打印截图进行核对 (如图 2 所示)。

## 2 制作方法

### 2.1 制作照片名对应的数据表

如图 2 所示。

	A	B	C	D	E
1	学号	姓名	身份证号		
2	0801	017	1302041992080		
3	0802	049	1302241992082		
4	0802	051	1302021993040		
5	0802	064	1302271991021		
6	0803	081	1302271992092		
7	0803	112	1302301993043		
8	0804	139	2321261993062		
9	0805	175	1302301992080		
10	0801	001	1302301993082		
11	0901	007	1130301993020		

图 2 数据表截图

将工作簿 sheet1 工作表命名为“数据”, 删除多余的工作表, 整个工作表的单元格数据格式设置为文本型。

### 2.2 程序结构

从“工具”→“宏”→“Visual Basic 编辑器” (快捷方式为 Alt + F11) 启动 VBE, 插入 3 个用户窗体和 5 个模块, 如图 3 所示。



图 3 编辑宏

UserForm1 用户窗体 (名称为【生成以身份证号码为文件名的照片】) 及各个控件名称, 如图 4 所示。





## DATABASE

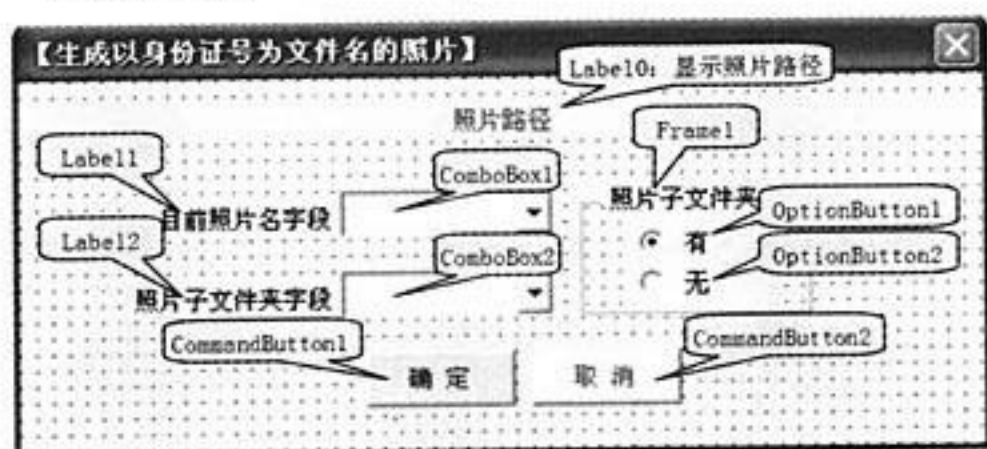


图4 UserForm1 界面

UserForm2 用户窗体 (名称为【从身份证号为文件名的照片中提取被标记的照片】) 及各个控件名称, 如图5所示。

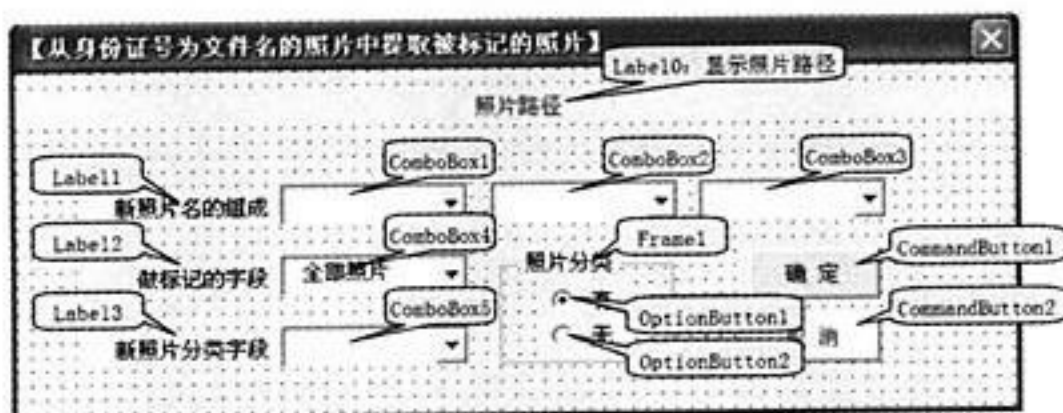


图5 UserForm2 界面

UserForm3 用户窗体 (名称为【根据需要重命名照片】) 及各个控件名称, 如图6所示。

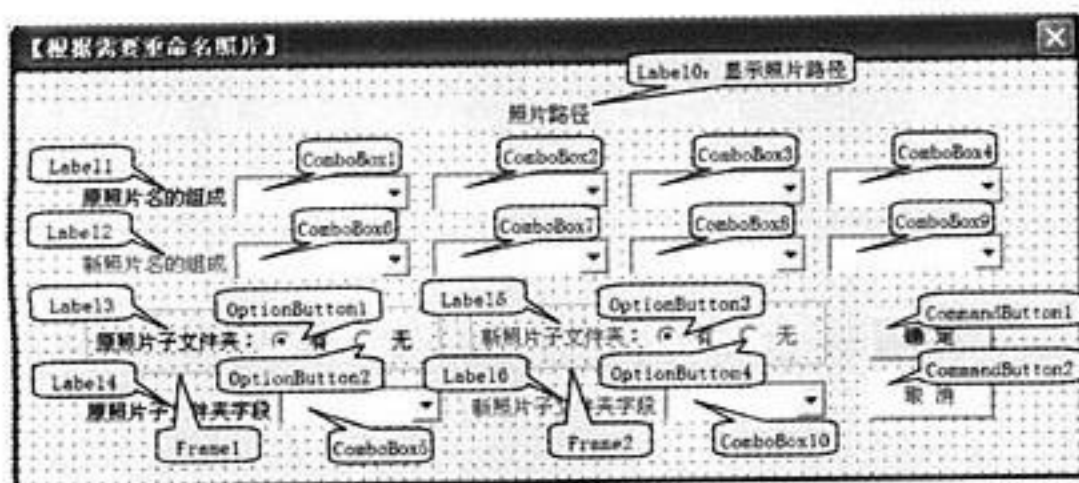


图6 UserForm3 界面

注意:

设置用户窗体的各下拉列表框的“Style”属性为: 2 - fmStyleDropDownList (只能下拉选取字段名, 不能键盘输入)。

设置3个用户窗体的单选按钮 OptionButton1 (有)、OptionButton3 (有) 的值为 True。

### 2.3 程序代码

(1) [ThisWorkbook] 中的代码:

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
'禁用标题栏【关闭】按钮
'声明变量
Dim BClose As Boolean
If BClose = False Then
Cancel = True
MsgBox "请使用系统提供的【关闭系统】命令关闭工作簿!", 48, "【提示】"
```

```
End If
End Sub
```

(2) [UserForm1] 窗体1中的代码:

```
Private Sub UserForm_QueryClose (Cancel As Integer, CloseMode As Integer) '禁用窗体【关闭】按钮
If CloseMode <> 1 Then Cancel = 1
```

```
End Sub
```

Private Sub OptionButton1\_Click() '【单选按钮1】代码

```
Me.Label2.Enabled = True
```

```
Me.ComboBox2.Enabled = True
```

```
End Sub
```

Private Sub OptionButton2\_Click() '【单选按钮2】代码

```
Me.Label2.Enabled = False
```

```
Me.ComboBox2.Enabled = False
```

```
End Sub
```

Private Sub CommandButton1\_Click() '【确定】按钮代码

```
'声明变量
```

```
Dim m As Integer
```

```
'选择原照片名字段
```

```
For m = 1 To [IV1].End(xlToLeft).Column
```

```
If Cells(1, m) = Me.ComboBox1.Text Then
```

```
Cells(3, 256) = m
```

```
Exit For
```

```
End If
```

```
Next
```

```
'判断是否选择了原照片名字段
```

```
If Me.ComboBox1.Text = "" Then
```

```
MsgBox "尚未选择【原照片名】字段。", 48, "【提示】"
```

```
Me.ComboBox1.SetFocus
```

```
Exit Sub
```

```
End If
```

```
'选择原照片所在的子文件夹字段
```

```
For m = 1 To [IV1].End(xlToLeft).Column
```

```
If Cells(1, m) = Me.ComboBox2.Text Then
```

```
Cells(4, 256) = m
```

```
Exit For
```

```
End If
```

```
Next
```

```
'判断是否选择了原照片所在的子文件夹字段
```

```
If Me.ComboBox2.Enabled = True And Me.ComboBox2.
```

```
Text = "" Then
```

```
MsgBox "尚未选择【照片所在的子文件夹】字段?", 48,
```

```
"【提示】"
```

```
Me.ComboBox2.SetFocus
```

```
Exit Sub
```

```
End If
```

```
Unload Me
```

```
End Sub
```

Private Sub CommandButton2\_Click() '【取消】按钮代码

```
Cells(5, 256) = "Q"
```

```
Unload Me
```

```
End Sub
```





(3) [UserForm2] 窗体 2 中的代码:

```
Private Sub UserForm_QueryClose (Cancel As Integer,
CloseMode As Integer) '禁用窗体【关闭】按钮
    If CloseMode <> 1 Then Cancel = 1
End Sub
Private Sub OptionButton1_Click() '【单选按钮 1】代码
    Me.Label3.Enabled = True
    Me.ComboBox5.Enabled = True
End Sub
Private Sub OptionButton2_Click() '【单选按钮 2】代码
    Me.Label3.Enabled = False
    Me.ComboBox5.Enabled = False
End Sub
Private Sub CommandButton1_Click() '【确定】代码
    '声明变量
    Dim m As Integer
    '读取新照片名
    For m = 1 To [IV1].End(xlToLeft).Column
        If Cells(1, m) = Me.ComboBox1.Text Then
            Cells(4, 256) = m
            Exit For
        End If
    Next
    For m = 1 To [IV1].End(xlToLeft).Column
        If Cells(1, m) = Me.ComboBox2.Text Then
            Cells(5, 256) = m
            Exit For
        End If
    Next
    For m = 1 To [IV1].End(xlToLeft).Column
        If Cells(1, m) = Me.ComboBox3.Text Then
            Cells(6, 256) = m
            Exit For
        End If
    Next
    '检查是否有组成新文件名的字段
    If Me.ComboBox1.Text & Me.ComboBox2.Text & Me.
ComboBox3.Text = "" Then
        MsgBox "尚未选择【新照片名】字段(最少选一个)。",
48, "【提示】"
        Me.ComboBox1.SetFocus
    End Sub
End If
'检查组成原文件名的字段是否出现重复
For n = 1 To 3
    For m = 1 To 3
        If m <> n Then
            If Cells(3 + n, 256) = Cells(3 + m, 256) And Cells(3 +
n, 256) <> "" Then
                MsgBox "组成【原文件名】的字段有重复,请核实。",
48, "【提示】"
                Me.Controls("ComboBox" & m).SetFocus
            End If
        End If
    Next
End Sub
```

```
Exit Sub
End If
End If
Next
Next
'读取标记字段
Cells(3, 256) = Cells(2, 256)
For m = 1 To [IV1].End(xlToLeft).Column
    If Cells(1, m) = Me.ComboBox4.Text Then
        Cells(3, 256) = m
        Exit For
    End If
Next
'读取照片分类字段
For m = 1 To [IV1].End(xlToLeft).Column
    If Cells(1, m) = Me.ComboBox5.Text Then
        Cells(7, 256) = m
        Exit For
    End If
Next
'判断是否选择了照片分类字段
If Me.ComboBox5.Enabled = True And Me.ComboBox5.
Text = "" Then
    MsgBox "尚未选择【新照片分类用字段】。", 48, "【提
示】"
    Me.ComboBox5.SetFocus
End Sub
End If
Unload Me
End Sub

Private Sub CommandButton2_Click() '【取消】代码
    Cells(8, 256) = "Q"
    Unload Me
End Sub
```

(4) [UserForm3] 窗体 3 中的代码:

```
Private Sub UserForm_QueryClose (Cancel As Integer,
CloseMode As Integer) '禁用窗体【关闭】按钮
    If CloseMode <> 1 Then Cancel = 1
End Sub

Private Sub OptionButton1_Click() '【单选按钮 1】代码
    Me.Label4.Enabled = True
    Me.ComboBox5.Enabled = True
End Sub

Private Sub OptionButton2_Click() '【单选按钮 2】代码
    Me.Label4.Enabled = False
    Me.ComboBox5.Enabled = False
End Sub
Private Sub OptionButton3_Click() '【单选按钮 3】代码
```





## DATABASE

```

Me.Label6.Enabled = True
Me.ComboBox10.Enabled = True
End Sub

Private Sub OptionButton4_Click() '【单选按钮 4】代码
    Me.Label6.Enabled = False
    Me.ComboBox10.Enabled = False
End Sub

Private Sub CommandButton1_Click() '【确定】按钮代码
    ' 声明变量
    Dim m As Integer
    ' 读取原照片名
    For m = 1 To [IV1].End(xlToLeft).Column
        If Cells(1, m) = Me.ComboBox1.Text Then
            Cells(3, 255) = m
            Exit For
        End If
    Next
    For m = 1 To [IV1].End(xlToLeft).Column
        If Cells(1, m) = Me.ComboBox2.Text Then
            Cells(4, 255) = m
            Exit For
        End If
    Next
    For m = 1 To [IV1].End(xlToLeft).Column
        If Cells(1, m) = Me.ComboBox3.Text Then
            Cells(5, 255) = m
            Exit For
        End If
    Next
    For m = 1 To [IV1].End(xlToLeft).Column
        If Cells(1, m) = Me.ComboBox4.Text Then
            Cells(6, 255) = m
            Exit For
        End If
    Next
    ' 判断是否有组成原文件名的字段
    If Me.ComboBox1.Text & Me.ComboBox2.Text & Me.
    ComboBox3.Text & Me.ComboBox4.Text = "" Then
        MsgBox "尚未选择【新照片名】字段(最少选一个)。",
48, "【提示】"
        Me.ComboBox1.SetFocus
        Exit Sub
    End If
    ' 检查组成原文件名的字段是否出现重复
    For n = 1 To 4
        For m = 1 To 4
            If m <> n Then
                If Cells(2 + n, 255) = Cells(2 + m, 255) And Cells(2 +
n, 255) <> "" Then
                    MsgBox "组成【原文件名】的字段有重复, 请核实。

```

```

", 48, "【提示】"
        Me.Controls("ComboBox" & m).SetFocus
        Exit Sub
    End If
End If
Next
Next
' 读取新照片名
For m = 1 To [IV1].End(xlToLeft).Column
    If Cells(1, m) = Me.ComboBox6.Text Then
        Cells(3, 256) = m
        Exit For
    End If
Next
For m = 1 To [IV1].End(xlToLeft).Column
    If Cells(1, m) = Me.ComboBox7.Text Then
        Cells(4, 256) = m
        Exit For
    End If
Next
For m = 1 To [IV1].End(xlToLeft).Column
    If Cells(1, m) = Me.ComboBox8.Text Then
        Cells(5, 256) = m
        Exit For
    End If
Next
For m = 1 To [IV1].End(xlToLeft).Column
    If Cells(1, m) = Me.ComboBox9.Text Then
        Cells(6, 256) = m
        Exit For
    End If
Next
' 判断是否有组成新文件名的字段
If Me.ComboBox6.Text & Me.ComboBox7.Text & Me.
ComboBox8.Text & Me.ComboBox9.Text = "" Then
    MsgBox "尚未选择【新照片名】字段(最少选一个)。",
48, "【提示】"
    Me.ComboBox6.SetFocus
    Exit Sub
End If
' 检查组成新文件名的字段是否出现重复
For n = 1 To 4
    For m = 1 To 4
        If m <> n Then
            If Cells(2 + n, 256) = Cells(2 + m, 256) And Cells(2 +
n, 256) <> "" Then
                MsgBox "组成【新文件名】的字段有重复, 请核实。
", 48, "【提示】"
        Me.Controls("ComboBox" & 5 + m).SetFocus
        Exit Sub
    End If
End If

```





```

Next
Next
' 读取原文件名子文件夹字段名
For m = 1 To [IV1].End(xlToLeft).Column
    If Cells(1, m) = Me.ComboBox5.Text Then
        Cells(2, 255) = m
        Exit For
    End If
Next
' 判断是否选择了原文件名子文件夹字段名
If Me.ComboBox5.Enabled = True And Me.ComboBox5.
Text = "" Then
    MsgBox " 尚未选择【原照片子文件字段】。", 48, "【提示】"
    Me.ComboBox5.SetFocus
    Exit Sub
End If
' 读取新文件名子文件夹字段名
For m = 1 To [IV1].End(xlToLeft).Column
    If Cells(1, m) = Me.ComboBox10.Text Then
        Cells(2, 256) = m
        Exit For
    End If
Next
' 判断是否选择了新文件名子文件夹字段名
If Me.ComboBox10.Enabled = True And Me.ComboBox10.
Text = "" Then
    MsgBox " 尚未选择【新照片子文件字段】。", 48, "【提示】"
    Me.ComboBox10.SetFocus
    Exit Sub
End If
Unload Me
End Sub

Private Sub CommandButton2_Click() '【取消】按钮代码
    Cells(7, 256) = "Q"
    Unload Me
End Sub

```

(5) [模块 1—启动程序] 中的代码:

```

Sub auto_open()
    Application.ScreenUpdating = False ' 关闭屏显
    On Error Resume Next
    ' 声明变量
    Dim i As Integer
    Dim NewMenu As CommandBarPopup
    Dim MenuItem As CommandBarControl
    Dim SubMenuItem As CommandBarButton
    ' 隐藏常规菜单项
    For i = 1 To Application.CommandBars(1).Controls.Count
        Application.CommandBars(1).Controls(i).Visible = False
    Next

```

```

' 如果菜单已存在,则删除该菜单
CommandBars(1).Controls("【导入信息】").Delete
CommandBars(1).Controls("【验证身份证号】").Delete
CommandBars(1).Controls("【处理照片名】").Delete
CommandBars(1).Controls("【关闭系统】").Delete
' 设置新菜单为临时的
Set NewMenu = CommandBars (1).Controls.Add (Type:=
msoControlPopup, Temporary:=True)
' 添加菜单标题
NewMenu.Caption = "【导入信息】"
' 添加菜单项
Set MenuItem = NewMenu.Controls.Add (Type:=
msoControlButton)
With MenuItem
    .BeginGroup = True
    .Caption = "导入信息"
    .FacId = 136
    .OnAction = "导入信息"
End With
' 设置新菜单为临时的
Set NewMenu = CommandBars (1).Controls.Add (Type:=
msoControlPopup, Temporary:=True)
' 添加菜单标题
NewMenu.Caption = "【验证身份证号】"
' 添加菜单项
Set MenuItem = NewMenu.Controls.Add (Type:=
msoControlButton)
With MenuItem
    .BeginGroup = True
    .Caption = "验证号码的正确性"
    .FacId = 202
    .OnAction = "验证正确性"
End With
' 设置新菜单为临时的
Set NewMenu = CommandBars (1).Controls.Add (Type:=
msoControlPopup, Temporary:=True)
' 添加菜单标题
NewMenu.Caption = "【处理照片名】"
' 添加菜单项
Set MenuItem = NewMenu.Controls.Add (Type:=
msoControlButton)
With MenuItem
    .BeginGroup = True
    .Caption = "生成以身份证号为文件名的照片"
    .FacId = 326
    .OnAction = "生成号照"
End With
' 添加菜单项
Set MenuItem = NewMenu.Controls.Add (Type:=
msoControlButton)
With MenuItem
    .BeginGroup = True

```





## DATABASE

```

.Caption = "从身份证号为文件名的照片中提取照片"
.FaceId = 326
.OnAction = "号照中提"
End With
' 添加菜单项
Set MenuItem = NewMenu.Controls.Add (Type: =
msoControlButton)
With MenuItem
.BeginGroup = True
.Caption = "按需要提取照片"
.FaceId = 326
.OnAction = "按需要提"
End With
' 设置新菜单为临时的
Set NewMenu = CommandBars (1).Controls.Add (Type: =
msoControlPopup, Temporary:=True)
' 添加菜单标题
NewMenu.Caption = "【关闭系统】"
' 添加菜单项
Set MenuItem = NewMenu.Controls.Add (Type: =
msoControlButton)
With MenuItem
.BeginGroup = True
.Caption = "关闭系统"
.FaceId = 1088
.OnAction = "关闭系统"
End With
Application.ScreenUpdating = True ' 恢复屏显
End Sub

```

(6) [模块 2—导入信息] 中的代码。

以前文章有介绍，不再赘述。

(7) 【模块 3—验正确性】中的代码。

以前文章有介绍，不再赘述。

(8) [模块 4—提取照片] 中的代码：

```

Sub 生成号照()
On Error Resume Next
' 声明变量
Dim h, i, l, m As Integer
h = [A65536].End(xlUp).Row
l = [IV1].End(xlToLeft).Column
Dim X ' X-为变体变量
' 清除数据和底色
Columns(Left(Cells(1, l + 1).Address(0, 0), Len(Cells(1, l +
1).Address(0, 0)) - 1) & ":IV").ClearContents
Rows("2:65536").Interior.ColorIndex = xlNone
' 确定身份证号码所在的列
For i = 1 To l
' 把小写字母转化为大写字母
Cells(2, i) = UCase(Cells(2, i))
If Len (Cells (2, i)) = 18 And InStr ("0123456789X", Right
(Cells(2, i), 1)) > 0 Then

```

```

For m = 1 To 17
If InStr("0123456789", Mid(Cells(2, i), m, 1)) = 0 Then
GoTo XYL: 'XYL=下一列
End If
Next
Cells(2, 256) = i
GoTo JX: 'JX=继续
End If
XYL: 'XYL=下一列
Next
MsgBox "没有身份证号数据,请核实。", 48, "【提示】"
Exit Sub
JX: 'JX=继续
' 创建文件夹
Set objShell = CreateObject("Shell.Application")
Set objFolder = objShell.BrowseForFolder(0, Chr(10) & "【请
选择照片所在的文件夹】", 0, 0)
If Not objFolder Is Nothing Then
WJJ = objFolder.self.Path
End If
Set objFolder = Nothing
Set objShell = Nothing
If WJJ = "" Or WJJ = False Then Exit Sub
If Right(WJJ, 1) <> "\" Then WJJ = WJJ & "\"
XWJJ = WJJ & "身份证号照片\"
Set fs = CreateObject("Scripting.FileSystemObject")
Set f = fs.GetFolder(XWJJ)
f.Delete
MkDir XWJJ
' 读取字段名
For m = 1 To [IV1].End(xlToLeft).Column
UserForm1.ComboBox1.AddItem Cells(1, m)
' 选择照片名字段
UserForm1.ComboBox2.AddItem Cells(1, m)
' 选择照片子文件夹字段
Next
' 读取照片路径
UserForm1.Label0.Caption = " 原路径为:[" & WJJ & "]" 新
路径为:[" & XWJJ & "]"
' 显示窗体
UserForm1.Show
If Cells(5, 256) = "Q" Then Exit Sub
' 提取照片并改名
For m = 2 To [A65536].End(xlUp).Row
If Cells (4, 256) <> "" Then YWJ = WJJ & Cells (m, Val
(Cells (4, 256))) & "\" & Cells (m, Val(Cells (3, 256))) & ".JPG"
YWJ=原文件
If Cells(4, 256) = "" Then YWJ = WJJ & Cells(m, Val(Cells
(3, 256))) & ".JPG" 'YWJ=原文件
If Dir(YWJ) <> "" Then
FileCopy YWJ, XWJJ & Cells(m, Val(Cells(2, 256))) & ".
JPG" ' 为身份证号

```





```

Else
    Range(Cells(m, 1), Cells(m, l)).Interior.ColorIndex = 36
    Cells(m, l + 1) = "缺照片"
End If
Next
'统计错误的记录个数
i = 0
For m = 2 To h
    If Cells(m, l + 1) <> "" Then i = i + 1
Next
MsgBox "应提取 " & h - 1 & " 张, 缺 " & i & " 张, 共提 " & h - 1 - i & " 张, 保存在 [" & WJJ & "身份证号照片" & "]" 中。", 48, "【提示】"
End Sub

Sub 号照中提()
    On Error Resume Next
    '声明变量
    Dim h, i, l, m As Integer
    h = [A65536].End(xlUp).Row
    l = [IV1].End(xlToLeft).Column
    '清除数据和底色
    Columns(Left(Cells(1, l + 1).Address(0, 0), Len(Cells(1, l + 1).Address(0, 0)) - 1) & ":IV").ClearContents
    Rows("2:65536").Interior.ColorIndex = xlNone
    '确定身份证号码所在的列
    For i = 1 To l
        '把小写字母转化为大写字母
        Cells(2, i) = UCase(Cells(2, i))
        If Len (Cells (2, i)) = 18 And InStr ("0123456789X", Right (Cells(2, i), 1)) > 0 Then
            For m = 1 To 17
                If InStr("0123456789", Mid(Cells(2, i), m, 1)) = 0 Then
                    GoTo XYL: 'XYL=下一列
                End If
            Next
            Cells(2, 256) = i
            GoTo JX: 'JX=继续
        End If
    XYL: 'XYL=下一列
    Next
    MsgBox "没有身份证号数据, 请核实。", 48, "【提示】"
    Exit Sub
    JX: 'JX=继续
    '创建文件夹
    Set objShell = CreateObject("Shell.Application")
    Set objFolder = objShell.BrowseForFolder(0, Chr(10) & "【请选择照片所在的文件夹】", 0, 0)
    If Not objFolder Is Nothing Then
        WJJ = objFolder.self.Path
    End If
    Set objFolder = Nothing

```

```

Set objShell = Nothing
If WJJ = "" Or WJJ = False Then Exit Sub
If Right(WJJ, 1) <> "\" Then WJJ = WJJ & "\"
XWJJ = WJJ & "提照片\"
Set fs = CreateObject("Scripting.FileSystemObject")
Set f = fs.GetFolder(XWJJ)
f.Delete
MkDir XWJJ
'读取字段名
For m = 1 To [IV1].End(xlToLeft).Column + 1
    UserForm2.ComboBox1.AddItem Cells(1, m)
    UserForm2.ComboBox2.AddItem Cells(1, m)
    UserForm2.ComboBox3.AddItem Cells(1, m)
    UserForm2.ComboBox4.AddItem Cells(1, m)
    UserForm2.ComboBox5.AddItem Cells(1, m)
Next
'读取照片路径
UserForm2.Label0.Caption = " 原路径为:[" & WJJ & "]" 新
路径为:[" & XWJJ & "]"
'显示窗体
UserForm2.Show
If Cells(8, 256) = "Q" Then Exit Sub
'创建照片分类子文件夹
If Cells(7, 256) <> "" Then
    For m = 2 To [A65536].End(xlUp).Row
        If Cells(m, Val(Cells(7, 256))) <> Cells(1 + m, Val(Cells(7, 256))) Then MkDir XWJJ & Cells(m, Val(Cells(7, 256)))
    Next
End If
'提取照片
For m = 2 To [A65536].End(xlUp).Row
    If Dir(WJJ & Cells(m, Val(Cells(2, 256))) & ".JPG") <> "" And Cells(m, Val(Cells(3, 256))) <> "" Then
        If Cells(7, 256) <> "" Then MBLJ = XWJJ & Cells(m, Val(Cells(7, 256))) & "\" 'MBLJ-目标路径
        If Cells(7, 256) = "" Then MBLJ = XWJJ
        'MBLJ-目标路径
        XWJM = ""
        For i = 1 To 3
            If Cells(3 + i, 256) <> "" Then
                XWJM = XWJM & Trim (Cells (m, Val (Cells (3 + i, 256)))) '新文件名
            End If
        Next
        FileCopy WJJ & Cells (m, Val (Cells (2, 256))) & ".JPG", MBLJ & XWJM & ".JPG"
    Else
        Range(Cells(m, 1), Cells(m, l)).Interior.ColorIndex = 36
        Cells(m, l + 1) = "缺照片"
    End If
Next
'统计缺照片数

```





## DATABASE

```

i = 0
For m = 2 To h
    If Cells(m, l + 1) <> "" Then i = i + 1
Next
MsgBox "应提取 " & h - 1 & " 张, 缺 " & i & " 张, 共提 " & h - 1 - i & " 张, 保存在 [" & XWJJ & "] 中。", 48, "【提示】"
End Sub

Sub 按需要提()
    On Error Resume Next
    ' 声明变量
    Dim h, i, l, m As Integer
    h = [A65536].End(xlUp).Row
    l = [IV1].End(xlToLeft).Column
    ' 清除数据和底色
    Columns(Left(Cells(1, l + 1).Address(0, 0), Len(Cells(1, l + 1).Address(0, 0)) - 1) & ":IV").ClearContents
    Rows("2:65536").Interior.ColorIndex = xlNone
    ' 创建文件夹
    Set objShell = CreateObject("Shell.Application")
    Set objFolder = objShell.BrowseForFolder(0, Chr(10) & "【请选择照片所在的文件夹】", 0, 0)
    If Not objFolder Is Nothing Then
        WJJ = objFolder.self.Path
    End If
    Set objFolder = Nothing
    Set objShell = Nothing
    If WJJ = "" Or WJJ = False Then Exit Sub
    If Right(WJJ, 1) <> "\" Then WJJ = WJJ & "\"
    XWJJ = WJJ & "提照片\"
    Set fs = CreateObject("Scripting.FileSystemObject")
    Set f = fs.GetFolder(XWJJ)
    f.Delete
    MkDir XWJJ
    ' 读取字段名
    For m = 1 To [IV1].End(xlToLeft).Column + 1
        UserForm3.ComboBox1.AddItem Cells(1, m)
        UserForm3.ComboBox2.AddItem Cells(1, m)
        UserForm3.ComboBox3.AddItem Cells(1, m)
        UserForm3.ComboBox4.AddItem Cells(1, m)
        UserForm3.ComboBox5.AddItem Cells(1, m)
        UserForm3.ComboBox6.AddItem Cells(1, m)
        UserForm3.ComboBox7.AddItem Cells(1, m)
        UserForm3.ComboBox8.AddItem Cells(1, m)
        UserForm3.ComboBox9.AddItem Cells(1, m)
        UserForm3.ComboBox10.AddItem Cells(1, m)
    Next
    ' 读取照片路径
    UserForm3.Label0.Caption = "原路径为: [" & WJJ & "] 新路径为: [" & XWJJ & "]"
    ' 显示窗体
    UserForm3.Show

```

```

If Cells(7, 256) = "Q" Then Exit Sub
' 创建照片分类子文件夹
If Cells(2, 256) <> "" Then
    For m = 2 To [A65536].End(xlUp).Row
        If Cells(m, Val(Cells(2, 256))) <> Cells(1 + m, Val(Cells(2, 256))) Then MkDir XWJJ & Cells(m, Val(Cells(2, 256)))
    Next
End If
' 提取照片
For m = 2 To [A65536].End(xlUp).Row
    ' 原文件名
    YWJM = ""
    For i = 1 To 4
        If Cells(2 + i, 255) <> "" Then
            YWJM = YWJM & Trim(Cells(m, Val(Cells(2 + i, 255))))
        ' YWJM-原文件名
    End If
    Next
    ' 新文件名
    XWJM = ""
    For i = 1 To 4
        If Cells(2 + i, 256) <> "" Then
            XWJM = XWJM & Trim(Cells(m, Val(Cells(2 + i, 256))))
        ' XWJM-新文件名
    End If
    Next
    If Cells(2, 255) <> "" Then
        If Cells(2, 256) <> "" Then
            If Dir(WJJ & Cells(m, Val(Cells(2, 255))) & "\" & YWJM & ".JPG") <> "" Then
                FileCopy WJJ & Cells(m, Val(Cells(2, 255))) & "\" & YWJM & ".JPG", XWJJ & Cells(m, Val(Cells(2, 256))) & "\" & XWJM & ".JPG"
            Else
                Range(Cells(m, 1), Cells(m, l)).Interior.ColorIndex = 36
                Cells(m, l + 1) = "缺照片"
            End If
        Else
            If Dir(WJJ & Cells(m, Val(Cells(2, 255))) & "\" & YWJM & ".JPG") <> "" Then
                FileCopy WJJ & Cells(m, Val(Cells(2, 255))) & "\" & YWJM & ".JPG", XWJJ & XWJM & ".JPG"
            Else
                Range(Cells(m, 1), Cells(m, l)).Interior.ColorIndex = 36
                Cells(m, l + 1) = "缺照片"
            End If
        End If
    Else
        If Cells(2, 256) <> "" Then
            If Dir(WJJ & YWJM & ".JPG") <> "" Then
                FileCopy WJJ & YWJM & ".JPG", XWJJ & Cells(m, Val(Cells(2, 256))) & "\" & XWJM & ".JPG"
            End If
        End If
    End If

```





```

Else
    Range(Cells(m, 1), Cells(m, l)).Interior.ColorIndex = 36
    Cells(m, l + 1) = "缺照片"
End If
Else
    If Dir(WJJ & "\" & YWJM & ".JPG") <> "" Then
        FileCopy WJJ & "\" & YWJM & ".JPG", XWJJ &
XWJM & ".JPG"
    Else
        Range(Cells(m, 1), Cells(m, l)).Interior.ColorIndex = 36
        Cells(m, l + 1) = "缺照片"
    End If
End If
End If
Next
'统计错误的记录个数
i = 0
For m = 2 To h
    If Cells(m, l + 1) <> "" Then i = i + 1
Next
MsgBox "应提取 " & h - 1 & " 张,缺 " & i & " 张,共提 " & h
- 1 - i & " 张,保存在 [' & XWJJ & '] 中。", 48, "[提示]"
End Sub

```

(9) [模块 5—关闭程序] 中的代码:

```

Sub 关闭系统()
    Application.ScreenUpdating = False '关闭屏显

```

(上接第 26 页)

```

sbit P05=P0^5; //用做程序运行指示灯
//上位机命令是 0x7F 为打开音响 0x7E 为关闭音响电源
//根据以上命令设 P02=0 为打开音响 P02=1 为关闭音响
...
//主程序
void main()
{ unsigned char chComm=0x00;
  P0M1=0x00;
  P0M2=0xFF; //设为推挽输出
  P02 = 1;
  P04 = 1;
  P05 = 0;
  Init_com(); //初始化串行口
  while(1)
  {
    if(RI)
    { RI=0;
      chComm=SBUF; //读取串行数据
      if(chComm==0x7F) //如果是
      {P02=0; //开启音响电源
        P04=0;} //电源指示灯亮
      else if(chComm==0x7E) //如果是
      {P02=1; //关闭音响电源

```

```

Application.WindowState = xlMaximized
'应用程序窗口处于"最大化"状态。
ActiveWindow.WindowState = xlMaximized
'文件窗口处于"最大化"状态。
Application.CommandBars(1).Enabled = True '显示"菜单栏"
Application.CommandBars("Standard").Visible = True
'显示"工具栏"
Application.CommandBars("Formatting").Visible = True
'显示"格式栏"
Application.DisplayFormulaBar = True '显示"编辑栏"
ActiveWindow.DisplayWorkbookTabs = True
'显示"工作表标签"
ActiveWindow.DisplayHeadings = True '显示"行号列标"
'显示菜单栏各项
Dim i As Long
For i = 1 To Application.CommandBars(1).Controls.Count
    Application.CommandBars(1).Controls(i).Visible = True
Next
Application.EnableEvents = False '禁用应用程序事件
Application.Quit '关闭
ActiveWorkbook.Close savechanges:=True '保存
Application.EnableEvents = True '恢复应用程序事件
Application.ScreenUpdating = True '打开屏显
End Sub

```

(收稿日期: 2012-11-19)

```

P04=1;}} //电源指示灯熄
P05=~P05; //程序运行指示灯工作
delay();
}}
//-----

```

## 3 结语

使用的程序范例选至应用程序《大厅音乐全自动播放器》的控制部分, 简单优秀一方面和点歌及播放程序没有关联, 只要计算机系统中有声音发出就可以捕获到音频信号的存在; 另一方面成本低廉, 选材方便容易实现。整个控制系统经过多年在酒家的实际应用, 性能稳定可靠。

说明: 选择的一并传到杂志社的应用程序《大厅音乐全自动播放器》包使用的编译器为 Borland C++ Builder6.0v。

## 参考文献

- [1] 李博轩. Visual C++ .NET 多媒体应用开发技术. 国防工业出版社, 2002.
- [2] 彭达, 王道智. Visual C++ 多媒体编程技术. 人民邮电出版社, 1999.

(收稿日期: 2012-11-06)





# ASP.NET 服务器日历控件的开发和使用

郭海伟

**摘要:** 通过一个具体的示例, 阐述开发 ASP.NET 服务器控件实施编程方法及其调试应用。

**关键词:** C# 语言; ASP.NET 技术; 控件

## 1 引言

ASP.NET 是目前应用最广泛的 Web 应用程序之一。Web 服务器控件是 ASP.NET 的一个重要组成部分, 其良好的封装和复用性为用户快速开发 Web 应用程序提供了支持。主要内容包括: (1) 新建一个 ASP.NET 服务器控件项目 (图 1 所示) 来编写 Web 服务器日历控件。(2) 再新建一个 ASP.NET Web 应用程序项目来调试和使用编写的 Web 服务器日历控件。

## 2 实施细则

开发平台是 Windows Server 2003+Visual Studio 2008, 开发语言 C#。

(1) 首先如图 1 新建了一个 ASP.NET 服务器控件项目, Visual Studio 自动产生如下的代码:

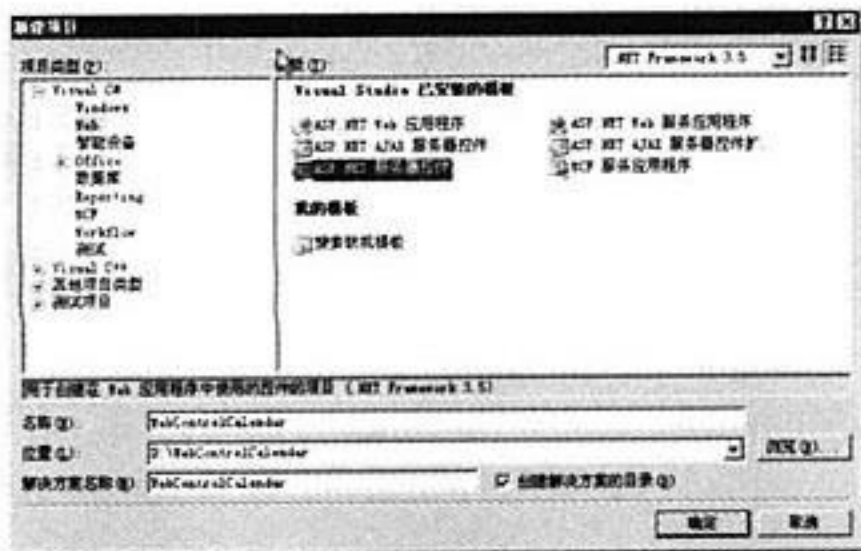


图 1 新建项目

```
namespace WebControlCalendar
{
    [DefaultProperty("Text")] //控件属性
    // ToolBoxData 是控件的定义。{0}表示控件的标记的前缀
    [ToolboxData ("<{0}:ServerControl1 runat=server></{0}:
    ServerControl1>")]
    public class ServerControl1 : WebControl
    {
        [Bindable(true)]
        [Category("Appearance")]
        [DefaultValue("")]
```

```
[Localizable(true)]
public string Text
{
    get
    {
        String s = (String)ViewState["Text"];
        return ((s == null) ? "[" + this.ID + "]" : s);
    }
    set
    {
        ViewState["Text"] = value;
    }
}
/// <summary>
/// 将此控件呈现给指定的输出参数。
/// </summary>
/// <param name="output"> 要写出到的 HTML 编写器
/// </param>
/// 编码工作就是改写 RenderContents(HtmlTextWriter
/// output)方法, 将 JavaScript 代码写入控件
protected override void RenderContents (HtmlTextWriter
output)
{
    protected override void Render(HtmlTextWriter output)
    {
        output.AddAttribute(HtmlTextWriterAttribute.Id, this.ID);
        output.AddAttribute(HtmlTextWriterAttribute.Name, this.ID);
        output.AddAttribute (HtmlTextWriterAttribute.Class,
this.CssClass);
        output.AddAttribute(HtmlTextWriterAttribute.Value, this.Text);
        output.AddAttribute (HtmlTextWriterAttribute.Style, "
cursor:hand");
        output.AddAttribute(HtmlTextWriterAttribute.Type, "text");
        output.AddAttribute(HtmlTextWriterAttribute.ReadOnly, "true");
        output.AddAttribute (HtmlTextWriterAttribute.
OnClick, "javascript:this.focus()");
        output.AddAttribute ("onFocus", "fPopCalendar (this,
this,PopCal); return false;");
        output.RenderBeginTag(HtmlTextWriterTag.Input);
        output.RenderEndTag();
```



```

if (! this.Page.IsClientScriptBlockRegistered("clientScript"))
{
    output.WriteLine("<script language='JavaScript'>");
    output.WriteLine("var gdCtrl = new Object();");
    output.WriteLine("var gcGray = \"#18758c\";");
    //非当前月应有日期字的颜色
    output.WriteLine("var gcToggle = \"highlight\";");
    //鼠标所在日期单元格的底色
    output.WriteLine("var gcBG = \"threedface\";");
    //日历背景色
    output.WriteLine("var gMonths = new Array(\"一
    月\", \"二月\", \"三月\", \"四月\", \"五月\", \"六月\", \"七月\", \"八月\",
    \"九月\", \"十月\", \"十一月\", \"十二月\");");
    output.WriteLine("var gdCurDate = new Date();");
    //gdCurDate 当前日期
    output.WriteLine("var giYear = gdCurDate.
    getFullYear();"); //giYear 当前年份
    output.WriteLine("var giMonth = gdCurDate.
    getMonth()+1;"); //giMonth 当前月份 (因为 getMonth()返
    //回的是 0-11 间的整数,故当前月要加 1)
    output.WriteLine("var giDay = gdCurDate.getDate
    ();"); //giDay 当前日
    output.WriteLine("var sxYear = giYear;"); //sxYear 所选年份
    output.WriteLine("var sxMonth = giMonth;");
    //sxMonth 所选月份
    output.WriteLine("var sxDay = giDay;");
    //sxDay 所选日
    output.WriteLine("var sxDatestr = gdCtrl.value;");
    //以前所选日期
    output.WriteLine("if (sxDatestr != \"\"){");
    output.WriteLine("var sxDate = new Date(sxDatestr);");
    output.WriteLine("sxYear = sxDate.getFullYear();");
    output.WriteLine("}");
    output.WriteLine("var VicPopCal = new Object();");
    #region 鼠标异动到某对象上的一系列函数
    output.WriteLine("function mouseover(obj){");
    output.WriteLine("obj.style.borderTop = 'buttonshadow
    1px solid';");
    output.WriteLine("obj.style.borderLeft = 'buttonshadow
    1px solid';");
    output.WriteLine("obj.style.borderRight = 'buttonhighlight
    1px solid';");
    output.WriteLine("obj.style.borderBottom = 'buttonhighlight
    1px solid';");
    output.WriteLine("}");
    output.WriteLine("function mouseout(obj){");
    output.WriteLine("obj.style.borderTop = 'buttonhighlight
    1px solid';");
    output.WriteLine("obj.style.borderLeft = 'buttonhighlight
    1px solid';");
    output.WriteLine("obj.style.borderRight = 'buttonshadow
    1px solid';");

```

```

    output.WriteLine("obj.style.borderBottom = 'buttonshadow
    1px solid';");
    output.WriteLine("}");
    output.WriteLine("function mousedown(obj){");
    output.WriteLine("obj.style.borderTop = 'buttonshadow
    1px solid';");
    output.WriteLine("obj.style.borderLeft = 'buttonshadow
    1px solid';");
    output.WriteLine("obj.style.borderRight = 'buttonhighlight
    1px solid';");
    output.WriteLine("obj.style.borderBottom = 'buttonhighlight
    1px solid';");
    output.WriteLine("}");
    output.WriteLine("function mouseup(obj){");
    output.WriteLine("obj.style.borderTop = 'buttonhighlight
    1px solid';");
    output.WriteLine("obj.style.borderLeft = 'buttonhighlight
    1px solid';");
    output.WriteLine("obj.style.borderRight = 'buttonshadow
    1px solid';");
    output.WriteLine("obj.style.borderBottom = 'buttonshadow
    1px solid';");
    output.WriteLine("}");
    #endregion
    #region 日历操作的一系列函数
    output.WriteLine("function fPopCalendar(popCtrl,
    dateCtrl, popCal){");
    output.WriteLine("parent.event.cancelBubble=true;");
    output.WriteLine("VicPopCal = popCal;");
    output.WriteLine("gdCtrl = dateCtrl;");
    output.WriteLine("fSetYearMon(giYear, giMonth);");
    output.WriteLine("var point = fGetXY(popCtrl);");
    output.WriteLine("with (VicPopCal.style) {left = point.x;
    top = point.y+popCtrl.offsetHeight+1;visibility = 'visible'}");
    output.WriteLine("VicPopCal.focus();");
    output.WriteLine("}");
    output.WriteLine("function fSetDate(iYear, iMonth, iDay){");
    output.WriteLine("if ((iYear == 0) && (iMonth ==
    0) && (iDay == 0)){");
    output.WriteLine("gdCtrl.value = \"\";");
    output.WriteLine("}");
    output.WriteLine("else{");
    output.WriteLine("if (iMonth < 10){iMonth = \"0\"+
    iMonth;"); //规格化时间
    output.WriteLine("if (iDay < 10){iDay = \"0\"+iDay;");
    output.WriteLine("gdCtrl.value = iYear +\"-\" +
    iMonth+\"-\"+iDay;");
    output.WriteLine("}");
    output.WriteLine("VicPopCal.style.visibility = 'hidden';");
    output.WriteLine("}");
    output.WriteLine("function fSetSelected(aCell){");
    output.WriteLine("var iOffset = 0;");

```





## NETWORK &amp; COMMUNICATION

```

output.WriteLine("var iYear = parseInt(document.
all.tbSelYear.value);");
output.WriteLine ("var iMonth = parseInt
(document.all.tbSelMonth.value);");
output.WriteLine("aCell.bgColor = gcBG;");
output.WriteLine("with (aCell.children['cellText']){");
output.WriteLine("var iDay = parseInt(innerText);");
output.WriteLine ("if (color==gcGray){iOffset =
(Victor<10)?-1:1;}");
output.WriteLine("iMonth += iOffset;");
output.WriteLine("if (iMonth<1) { iYear -- ;
iMonth = 12;}else{if (iMonth>12){iYear++;iMonth = 1;}}");
output.WriteLine("");
output.WriteLine("fSetDate(iYear, iMonth, iDay);");
output.WriteLine("");
output.WriteLine("function Point(iX, iY){this.x = iX;
this.y = iY;}");
output.WriteLine("function fBuildCal(iYear, iMonth){");
output.WriteLine("var aMonth=new Array();");
output.WriteLine("for(i=1;i<7;i++){aMonth[i]=new Array(i);}");
output.WriteLine ("var dCalDate=new Date(iYear,
iMonth-1, 1);");
output.WriteLine("var iDayOfFirst=dCalDate.getDay();");
output.WriteLine ("var iDaysInMonth=new Date
(iYear, iMonth, 0).getDate();");
output.WriteLine ("var iOffsetLast =new Date
(iYear, iMonth-1, 0).getDate()-iDayOfFirst+1;");
output.WriteLine("var iDate = 1;");
output.WriteLine("var iNext = 1;");
output.WriteLine("for (d = 0; d < 7; d++){aMonth
[1][d] = (d<iDayOfFirst)?-(iOffsetLast+d):iDate++;}");
output.WriteLine("for (w = 2; w < 7; w++){for (d
= 0; d < 7; d++){aMonth [w][d] = (iDate<=iDaysInMonth)?
iDate++:-iNext++;}}");
output.WriteLine("return aMonth;");
output.WriteLine("");
output.WriteLine ("function fDrawCal (iYear,
iMonth, iDay, iCellWidth, iDateTextSize) {");
output.WriteLine("var WeekDay = new Array("日
", "一", "二", "三", "四", "五", "六");");
output.WriteLine ("var styleTD = " bgcolor=" +gcBG + "
width=" +iCellWidth + " bordercolor=" +gcBG + " valign =
'middle' align='center' style='font-size: 12px;background:
buttonface;border-top: buttonhighlight 1px solid;border-left:
buttonhighlight 1px solid;border-right: buttonshadow 1px
solid;border-bottom: buttonshadow 1px solid;";");
output.WriteLine("with (document) {");
output.WriteLine("write("<tr align='center'>");");
output.WriteLine("for(i=0; i<7; i++){write("<td height='20'
"+styleTD+"color:#18758c'>" + WeekDay[i] + "</td>");");
output.WriteLine("write("</tr>");");
output.WriteLine("for (w = 1; w < 7; w++) {");

```

```

output.WriteLine("write("<tr align='center'>");");
output.WriteLine("for (d = 0; d < 7; d++) {");
output.WriteLine ("write ("<td width='10%'
height='15' id =calCell " +styleTD + "cursor:hand;
onmouseover='mouseover (this)' onmouseout='mouseout
(this)' onmousedown='mousedown (this)' onmouseup =
'mouseup(this)' onclick='fSetSelected(this)'>");");
output.WriteLine ("write ("<font style='font-size:
13px;' id=cellText Victor='Liming Weng'> </font>");");
output.WriteLine("write("</td>");");
output.WriteLine("");
output.WriteLine("write("</tr>");");
output.WriteLine("");
output.WriteLine("");
output.WriteLine("");
output.WriteLine("function fUpdateCal(iYear, iMonth) {");
output.WriteLine("sxYear = iYear;");
output.WriteLine("sxMonth = iMonth;");
output.WriteLine("yeard1.innerText = sxYear + "年";");
output.WriteLine("monthtd1.innerText = gMonths[sxMonth-1];");
output.WriteLine("myMonth = fBuildCal(iYear, iMonth);");
output.WriteLine("var i = 0;");
output.WriteLine("for (w = 0; w < 6; w++){");
output.WriteLine("for (d = 0; d < 7; d++){");
output.WriteLine("with (cellText[(7*w)+d]) {");
output.WriteLine("Victor = i++;");
output.WriteLine("if (myMonth[w+1][d]<0) {");
output.WriteLine("color = gcGray;");
output.WriteLine("innerText = -myMonth[w+1][d];");
output.WriteLine("}else{");
output.WriteLine("color = ((d==0)||d==6)?'red':'black';");
output.WriteLine("innerText = myMonth[w+1][d];");
output.WriteLine("");
output.WriteLine("");
output.WriteLine("");
output.WriteLine("");
output.WriteLine("");
//设置列表框中的年份和月份
output.WriteLine("function fSetYearMon(iYear, iMon){");
output.WriteLine("sxYear = iYear;");
output.WriteLine("sxMonth = iMon;");
output.WriteLine("yeard1.innerText = sxYear + "年";");
output.WriteLine("monthtd1.innerText = gMonths[sxMonth-1];");
output.WriteLine ("document.all.tbSelMonth.
options[iMon-1].selected = true;");
output.WriteLine ("for (i = 0; i < document.all.
tbSelYear.length; i++){");
output.WriteLine ("if (document.all.tbSelYear.
options[i].value == iYear){");
output.WriteLine ("document.all.tbSelYear.options
[i].selected = true;");
output.WriteLine("");

```





```

output.WriteLine("}");
output.WriteLine("fUpdateCal(iYear, iMon);");
output.WriteLine("}");
output.WriteLine("function fPrevMonth(){");
    output.WriteLine ("var iMon = document.all.
tbSelMonth.value;");
    output.WriteLine ("var iYear = document.all.
tbSelYear.value;");
    output.WriteLine("if (--iMon<1) {");
    output.WriteLine("iMon = 12;");
    output.WriteLine("iYear--;");
    output.WriteLine("}");
    output.WriteLine("fSetYearMon(iYear, iMon);");
    output.WriteLine("}");
    output.WriteLine("function fNextMonth(){");
    output.WriteLine ("var iMon = document.all.
tbSelMonth.value;");
    output.WriteLine ("var iYear = document.all.
tbSelYear.value;");
    output.WriteLine("if (++iMon>12) {");
    output.WriteLine("iMon = 1;");
    output.WriteLine("iYear++;");
    output.WriteLine("}");
    output.WriteLine("fSetYearMon(iYear, iMon);");
    output.WriteLine("}");
    output.WriteLine("function fGetXY(aTag){");
    output.WriteLine("var oTmp = aTag;");
    output.WriteLine("var pt = new Point(0,0);");
    output.WriteLine("do {");
    output.WriteLine("pt.x += oTmp.offsetLeft;");
    output.WriteLine("pt.y += oTmp.offsetTop;");
    output.WriteLine("oTmp = oTmp.offsetParent;");
    output.WriteLine("} while(oTmp.tagName! =\"BODY\");");
    output.WriteLine("return pt;");
    output.WriteLine("}");
    #endregion
    #region 日历体
    output.WriteLine("with (document){");
    output.WriteLine ("write (\"<Div id='PopCal'
onclick='event.cancelBubble=true' style='POSITION:
absolute; VISIBILITY: hidden; bordercolor:#000000;border:2px
ridge;width:10;z-index:100;'>\");");
    output.WriteLine ("write (\"<table id='popTable'
border='1' bgcolor='#23f4f2' cellpadding='0' cellspacing='0'
style='font-size:12px'>\");");
    output.WriteLine("write(\"<TR>\");");
    output.WriteLine ("write (\"<td valign='middle'
align='center' style='cursor:default'>\");");
    //日历头
    output.WriteLine ("write (\"<table width='176'
border='0' cellpadding='0' cellspacing='0'>\");");
    output.WriteLine("write(\"<tr align='center'>\");");

```

```

//上一个月
    output.WriteLine("write(\"<td height='22' width=
'20' name='PrevMonth' style='font-family:webdings;
font-size:15px' onClick='fPrevMonth ()' onmouseover='this.
style.color=\\\"#ff9900\\\"' onmouseout='this.style.color=\\\"\\\"
'>3</td>\");");
    //显示和选择年份-----
    //显示年份
    output.WriteLine ("write (\"<td width='64' id =
'yeartd1' style='font-size:12px' onmouseover='yeartd1.style.
display=\\\"none\\\";yeartd2.style.display=\\\"\\\";' onmouseout=
'this.style.background=\\\"\\\">\");");
    output.WriteLine("write(sxYear + \"年 \");");
    output.WriteLine("write(\"</td>\");");
    //年份选择
    output.WriteLine ("write (\"<td width='64' id =
'yeartd2' style='display:none' onmouseout='yeartd2.style.
display=\\\"none\\\";yeartd1.style.display=\\\"\\\";'>\");");
    output.WriteLine("write(\"<SELECT style='width:
64px;font-size: 12px;font-family: 宋体;' id='tbSelYear'
onChange='fUpdateCal(document.all.tbSelYear.value, docum
ent.all.tbSelMonth.value);yeartd2.style.display=\\\"none\\\";yea
rtd1.style.display=\\\"\\\"; Victor='Won'>\");");
    output.WriteLine("for(i=1930;i<2015;i++){");
    output.WriteLine("write(\"<OPTION value='\"+i+\"
'>\"+i+\"年</OPTION>\");");
    output.WriteLine("}");
    output.WriteLine("write(\"</SELECT>\");");
    output.WriteLine("write(\"</td>\");");
    //显示和选择月份-----
    //显示月份
    output.WriteLine ("write (\"<td width='72' id =
'monthtd1' style='font-size:12px' onmouseover='monthtd1.
style.display=\\\"none\\\";monthtd2.style.display=\\\"\\\";'onmou
seout='this.style.background=\\\"\\\">\");");
    output.WriteLine("write(gMonths[sxMonth-1]);");
    output.WriteLine("write(\"</td>\");");
    //月份选择
    output.WriteLine ("write (\"<td width='72' id =
'monthtd2' style='display:none' onmouseout='monthtd2.
style.display=\\\"none\\\";monthtd1.style.display=\\\"\\\";'>\");");
    output.WriteLine ("write (\"<select style='width:
72px;font-size: 12px;font-family: 宋体;' id='tbSelMonth'
onChange='fUpdateCal(document.all.tbSelYear.value, docum
ent.all.tbSelMonth.value);monthtd2.style.display=\\\"none\\\";
monthtd1.style.display=\\\"\\\"; Victor='Won'>\");");
    output.WriteLine("for (i=0; i<12; i++){");
    output.WriteLine("write(\"<option value='\"+(i+1)+
\">\"+gMonths[i]+\"</option>\");");
    output.WriteLine("}");
    output.WriteLine("write(\"</SELECT>\");");
    output.WriteLine("write(\"</td>\");");

```





## NETWORK & COMMUNICATION

//下一个月

```
output.WriteLine("write(<td width='20' name='PrevMonth'
style='font-family: \\\"webdings\\\"; font-size: 15px' onclick =
'fNextMonth ()' onmouseover='this.style.color=\\\"#ff9900\\\"'
onmouseout='this.style.color=\\\"\\\">4</td>\\');");
```

```
output.WriteLine("write(</tr>\\');");
```

```
output.WriteLine("write(</table>\\');");
```

//-----日历头结束-----

```
output.WriteLine ("write (</td></TR><TR><td
align='center'>\\');");
```

```
output.WriteLine ("write (<DIV style =
'background-color: teal;' ><table width='100%' border='0'
bgcolor='threedface' cellpadding='0' cellspacing='0'>\\');");
output.WriteLine("fDrawCal(giYear, giMonth, giDay, 19, 14);");
output.WriteLine ("write (</table></
DIV>\\');");
```

```
output.WriteLine ("write (</td></TR><TR><TD
height='20' align='center' valign='bottom'>\\');");
```

```
output.WriteLine ("write (</TD></TR></TD></
TR></TABLE>\\');");
```

```
output.WriteLine("write(</Div>\\');");
```

```
output.WriteLine(");");
```

```
#endregion
```

```
output.WriteLine("</script>");
```

```
output.WriteLine("<SCRIPT event=onclick() for=
document>PopCal.style.visibility = 'hidden';</SCRIPT>");
```

```
this.Page.RegisterClientScriptBlock("clientScript", "");
```

```
}
```

```
}
```

```
}
```

然后如图 2 所示重新生成控件项目，会在工程 bin\debug 目录下成功生成 WebControlCalendar.dll 文件，完成 ASP.NET 服务器日历控件的编写。

(2) 图 3 所示新建一个 ASP.NET Web 应用程序项目来调试和使用前面开发完成的 ASP.NET 服务器日历控件。



图 2 生成控件

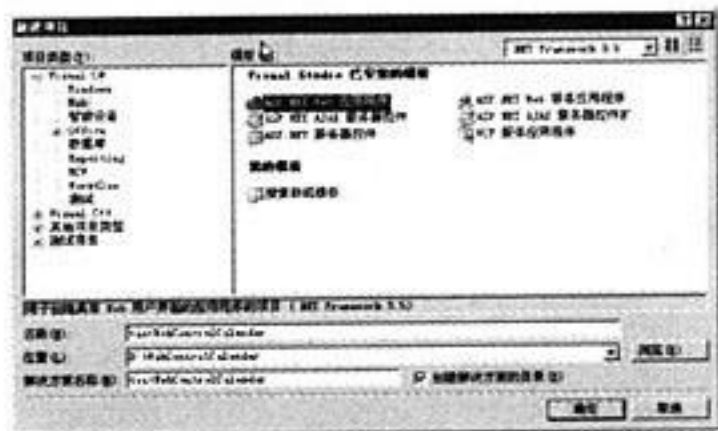


图 3 新建应用程序项目

首先通过如图 4 所示那样在工具栏通过添加选择项的方式来添加编译生成的 WebControlCalendar.dll 文件。

添加成功后会在工具箱中多出来一个选项，这样就成功地在 ASP.NET Web 应用程序中引用刚才开发完成的 ASP.NET 服务器日历控件，如图 5 所示。



图 4 添加选择项

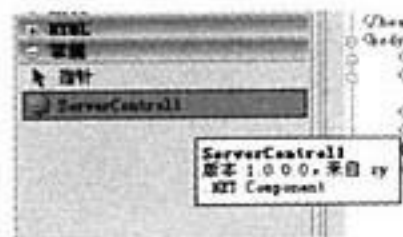


图 5 成功添加控件

现在可以直接把新生成的 ServerControl1 控件拖到新建的 Web 窗体上，切换到源视图状态，可以看到自动生成代码如下：  
`<webCalendar:ServerControl1 ID="ServerControl1" runat="server" />`

同时，在页面上部可以看到如下的标记：

```
<%@ Register Assembly="WebControlCalendar" Namespace="
WebControlCalendar" TagPrefix="webCalendar" %>
```

最后启动调试后的效果如图 6 所示。



图 6 调试效果

### 3 结语

为了提高软件的开发质量和效率，快速开发 Web 应用程序，采用封装算法、中间组件、服务器控件等很多手段来实现。本例抛砖引玉，希望对广大程序开发人员有所帮助。

### 参考文献

- [1] Jeffrey Richter, Applied Microsoft .NET Framework Programming, 世界图书出版公司, 2002, 09.
- [2] Dr.shahram Khosravi ASP.NET 2.0 服务器控件与组件开发高级编程, 郝刚, 田亮君, 陈文, 译, 人民邮电出版社, 2007, 06.
- [3] 范盛荣, VISUAL C# 2008 控件使用范例详解, 清华大学出版社, 2009, 04.

(收稿日期: 2012-09-19)



## JavaScript、ASP 车辆调度系统

商 涛

**摘 要：**应用 JavaScript、ASP、Access，并结合现代网络办公的需求，利用目前中小型企业现有局域网资源，采用 B/S 三层体系结构模式，开发一套《车辆调度系统》，该系统的实现和应用，不但提高办公效益，还省去了填写纸质用车申请单劳动强度，极大降低办公经费，提高办公效率，达到节能降耗目的。

**关键词：**JavaScript 脚本；ASP 技术；系统开发

## 1 引言

开发一套完整的应用软件，至少要精通 2 到 3 门计算机高级语言及部分数据库知识。无论是系统的数据“录入、查询、修改、删除、报表打印”等常见功能，都离不开数据库的后台支持。对开发程序有几点要求：（1）调研；（2）审核；（3）认定（定稿）；（4）绘制 N-S 流程图；（5）建数据库；（6）建表；（7）建表关联等。以下主要讲述使用 JavaScript、ASP、Access 开发车辆调度系统的关键步骤，并附核心代码。

## 2 数据库及表

车辆调度系统数据库是用 Access 创建的，名为 cldd.mdb，为了提高数据库的安全性，还可将扩展名 mdb 改为 Asp。数据库中主要包含 admin（超级管理员）表、ddjl（调度记录）表、yesqd（用车申请单）表，如图 1~图 3 所示。

admin : 表		
	字段名称	数据类型
	ID	自动编号
	admin	文本
	password	文本

图 1 超级管理员表

ddjl : 表			
	字段名称	数据类型	
	ID	自动编号	序号
	bm	文本	派车部门
	rq	日期/时间	调度日期
	xq	文本	星期
	xm	文本	驾驶员姓名
	gznr	文本	工作内容
	cph	文本	车牌号
	ycbm	文本	用车部门
	ccrq	日期/时间	发车日期
	ccsj	文本	发车时间
	hlsj	日期/时间	回来日期
	hlsj	文本	回来时间
	bz	备注	备注信息

图 2 车辆调度记录表

yesqd : 表			
	字段名称	数据类型	
	ID	自动编号	序号
	ck	文本	车辆类型
	ycsj	日期/时间	用车日期
	sj	文本	用车时间
	rs	数字	乘车人数
	qcd	文本	起始点
	qsj	日期/时间	出发日期
	rsj	日期/时间	返回日期
	yl	备注	车辆用途
	bz	备注	用车人联系
	bzl	备注	派车信息
	ycbm	文本	用车部门
	hrr	文本	部门负责人
	qtsj	文本	部门负责人时间
	pcyj	文本	派车信息
	pcsj	文本	派车时间

图 3 用车申请单表

## 3 核心代码

## 3.1 链接数据库 (conn.asp)

```
<%
Set conn = Server.CreateObject("ADODB.Connection")
conn.Open"driver={MicrosoftAccessDriver (*.mdb)};dbq = "&
server.MapPath("cldd.asp")' 打开数据库文件
set rs= server.createobject("adodb.recordset") ' 创建对象记
录指针
%>
```

## 3.2 主界面 (index.asp)

在该页面中采用 Frameset 框架元素：

```
<FRAMESET border=0
frameSpacing =0 rows =64,* frameBorder =no ><FRAME
name=top
src="top.asp" noResize scrolling=no>' 在 top 框架中调用 top.
asp 文件
<FRAMESET border=0
frameSpacing =0 frameBorder =no cols =150,* ><FRAME
name=left
src="left.asp" noResize scrolling=no>' 在 left 框架中调用 left.
asp 文件
```





## NETWORK & COMMUNICATION

```
<FRAME name=right
src="right.asp" noResize scrolling=yes>' 在 right 框架中调用
'right.asp 文件
</FRAMESET><noframes></noframes>
```

### 3.3 数据录入代码

数据录入代码就是数据添加程序，开发企业内部车辆调度系统，为了提升系统运行速度，本系统就不涉及到身份验证机制，但要用到更多的函数和变量，数据添加程序核心代码如下。

```
<! --#include file="conn.asp"-->' 调用数据库链接文件
<! --#include file="sysdate.asp"-->' 调用万年历文件
<%
Response.Buffer = True ' 打开缓冲
Dim cx,ycsj,sj ,rs ,qcd,qsj,zsj,zt,bz,bz1,ycbm,hzr,qzsj,pcyj,
pcsj ' 定义变量
' 获取文本框的值赋给变量
cx = Request.Form("cx ")
ycsj = Cstr( Request.Form("ycsj"))
sj = Cstr( Request.Form("sj"))
rs = Request.Form("rs ")
qcd = Request.Form("qcd ")
qsj = Cstr( Request.Form("qsj"))
zsj = Cstr( Request.Form("zsj"))
yt = Request.Form("yt ")
bz = Request.Form("bz ")
bz1 = Request.Form("bz1")
ycbm = Request.Form("ycbm ")
hzr = Request.Form("hzr ")
qzsj = Cstr( Request.Form("qzsj"))
pcyj = Request.Form("pcyj")
pcsj = Cstr( Request.Form("pcsj"))
%>
<%
set rs=server.createobject("adodb.recordset")
sqlstr="select * from ycsqd" ' 打开数据表
rs.open sqlstr,conn,1,3
sql="insert into ycsqd (cx,ycsj,sj ,rs ,qcd,qsj,zsj,zt,bz,
bz1,ycbm,hzr,qzsj,pcyj,pcsj)value("+cx+", "+ycsj+", "+sj+",
"+rs+", "+qcd+", "+qsj+", "+zsj+", "+yt+", "+bz+", "+
bz1+", "+ycbm+", "+hzr+", "+qzsj+", "+pcyj+", "+
pcsj+")" ' 向数据表中添加数据
conn.execute sql conn.close
set conn=nothing
response.write"<script>alert('录入成功!');back(0) </script>"
' 提示信息录入成功
end if
end if
%>
```

### 3.4 功能模块

只有部门领导正确登录系统后才能进入签字页，登录时要

输入正确的用户名、密码和部门名称才能成功登录，正确登录后才能进入自己的待办工作页面，单击待办工作标题提示符，即可进入最新单子签字页面完成签字工作。

部门领导登录代码：

```
<! --#include file="conn.asp"-->' 调用数据库
<%
dim user, pwd,bmmc ' 定义变量
user=trim(request.form("user")) ' 获取变量值
pwd=trim(request.form("pwd"))
bmmc=trim(request.form("bmmc"))
if user="" then
%>
<script language=javascript>
history.back()
alert("用户名不正确！")
</script>
<% end if %>
<%
if pwd="" then
%>
<script language=javascript>
history.back()
alert("密码不正确！")
</script>
<% end if %>
<%
mysql="select * from cd where user=" & replace(user,
";","'") & " and pwd = " & replace (pwd,";", "'") & " and bmmc = "
& replace(bmmc,";", "'") & ""
set rs=server.createobject("adodb.recordset")
rs.open mysql,conn,1,2
if rs.eof then
%>
<script language=javascript>
history.back()
alert("用户名或密码错误！")
</script>
<% else %>
<%
session("user")=user
response.redirect "pcqz3.asp" ' 进入签字页面
response.end
end if
%>
签字代码如下：
<%
dim user1,qzsj
qzsj=now() ' 自动获取当前系统时间
user1=session("user")
%>
<input class="red" readonly="flase" type="text" name="
```



```
x_hzr" id="x_hzr" size="6" maxlength="6" value="<%= Server.HTMLEncode(user1&"') %>">
```

```
<input class="red" readonly="flase" type="text"
name="x_qzs" id="x_qzs" size="18" maxlength="10" value="
<%= Server.HTMLEncode(qzs&"') %>">
```

查询调度记录:

本系统的调度记录查询分三种方式, 第一按用车用户名和用车年月进行查询, 方便查询某职工在某年某月内的用车量; 第二按出车日期查询, 方便查询某天的车辆调度情况; 第三按时间段来查询某一时段的出车情况。代码如下:

```
<! --#include file="conn.asp"--> ' 打开数据库链接文件
<! --#include file="sysdate.asp"--> ' 打开万年历文件
<%
    dim user,rq1 ' 定义变量
    user=trim(request("name")) ' 获取传送的变量值
    rq1=trim(request("year"))&"-"&trim(request("month"))
    set rs=server.createobject("adodb.recordset")
    sql="select * from ddjl where xm="&user&" and rq like
'%"&rq1&"%" ' 第一种查询方式
    sql="select * from ddjl where rq like #"&rq1&"#" '
第二种查询方式
    sql="select * from ddjl where rq between #"&rq1&"# and
#"&rq2&"#" ' 第三种查询方式
    rs.open sql,conn,1,2
    if rs.bof and rs.eof then
        response.write "<script>alert (' 没有找到你要查询的记录!
');history.back()</script> " response.end
    end if
    session("bh")=rs("ID") ' 把 ID 记录值赋给 Session 变量
```

(上接第 37 页)

```
t := (t1*tcrec.nSIZE)/iPSize;
ilInterval:=NUMTODSINTERVAL(t, 'SECOND');
else --如果未找到取表中区间, 则以配置表中的间隔作为下
--一个时间间隔
    NULL;
end if;
return (iResult);
EXCEPTION
WHEN OTHERS
THEN
    iResult:=SQLCODE;
    writelog ('calculateinterval',sql_stmt || ' ' || SUBSTR
(SQLERRM, 1, 300));
    return (iResult);
end calculateinterval;
```

## 6 结语

在实际自动售检票系统中使用 Oracle 数据库, 由于每条线

```
rs.pagesize=40 ' 当前页面最多显示 40 条记录
if request("page")<>" then
    epage=cint(request("page"))
    if epage<1 then epage=1
    if epage>rs.pagecount then epage=rs.pagecount
else
    epage=1
end if
rs.absolutepage=epage
%>
...
<%
for i=0 to rs.pagesize-1
if rs.bof or rs.eof then exit for
%>
...
<%
rs.movenext() ' 下一条
next
%>
...
<%
conn.close() ' 关闭数据库的链接
%>
```

## 4 结语

本系统是方便、快速地编写 B/S 三层体系结构模式的软件应用系统。经推广使用, 完全满足车辆调度管理的需要。

(收稿日期: 2012-09-07)

路的客流非常大, 故 Oracle 数据库中有一些表的交易记录非常多, 于是这些表都进行了分区, 在没有编写本文的程序包之前, 如果不及时增加分区会导致交易数据不能入库的错误, 故都是在每年年底编写一系列增加下一年每个月分区的 SQL 维护脚本, 人工执行这些脚本, 这样会导致有很多空间空着不能使用, 也容易发生错误, 通过编写程序包, 在经过多轮严密的测试和修改, 满足了上线运行的条件。

## 参考文献

- [1] Oracle. Oracle® Database VLDB and Partitioning Guide [M]. Oracle Press, 2011.
- [2] 谷长勇, 吴逸云, 单永红. ORACLE 11g 权威指南 [M]. 北京: 电子工业出版社, 2011.

(收稿日期: 2012-12-04)





## 邮件查看软件设计

刘步林

**摘要:** 邮件查阅软件的设计, 展示了现代企业项目经常用到的代码重用功能、编程风格、变量、窗体、类的命名规范以及指针变量在编程中的灵活应用。

**关键词:** 邮件; 软件复用; 公共单元; 控件

### 1 引言

邮件软件是目前 Internet 上使用最多的服务之一。电子邮件服务在信息的传播、交流等方面的应用越来越广泛, 它可以传递文字、图表、声音、电影等各类信息, 在现实生活中, 发送邮件, 查看邮件非常频繁, 工作上许多会议通知, 问题解决方案都是通过邮件联系, 邮件与我们生活密切相关。它具备简单、快捷、通信方便的特点。

平时都是通过其他软件公司开发的比较成熟的邮件查看工具如 Office 的 OUTLOOK、IE 或 hotmail 来查看自己的邮箱信息, 那如何开发研制属于自己个性化的邮箱查看工具, 符合自己的爱好, 配置自己喜欢的背景等, 这是一件非常有意义的事情。

### 2 需求分析

邮件查看软件的要求具有以下功能:

- (1) 建立邮箱目录。
- (2) 查看邮箱信件列表。
- (3) 查看某封邮件的具体内容。
- (4) 新建、修改、删除邮箱帐户信息。
- (5) 从 XML 文件中导入邮箱帐户信息。
- (6) 将树形目录邮箱帐户列表保存为 XML 文件。

### 3 软件设计

#### 3.1 软件功能结构

软件功能如图 1 所示。

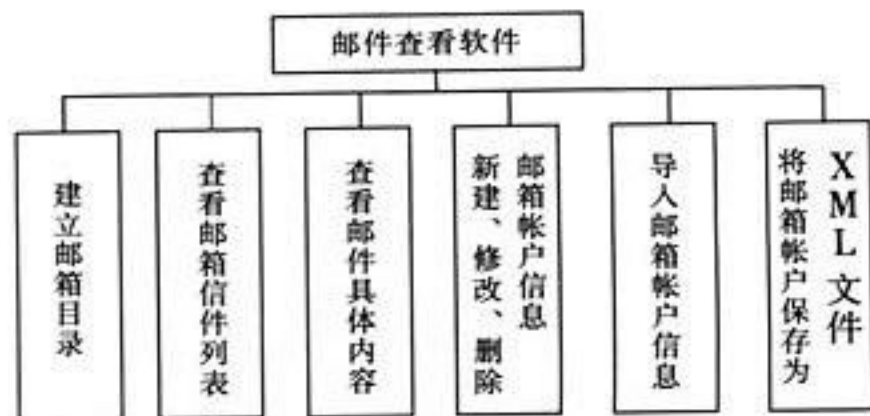


图 1 软件功能结构

#### 3.2 邮件处理流程

首先启动邮件查看软件, 建立邮箱帐户输入邮箱服务器名称、用户名、密码和端口号等, 然后连接测试是否成功, 如果不成功则检查是否输入有误, 重新配置邮箱帐户信息, 若连接成功, 则开始接收邮件, 查看邮箱信件列表, 查看邮件具体内容等。邮件处理流程如图 2 所示。

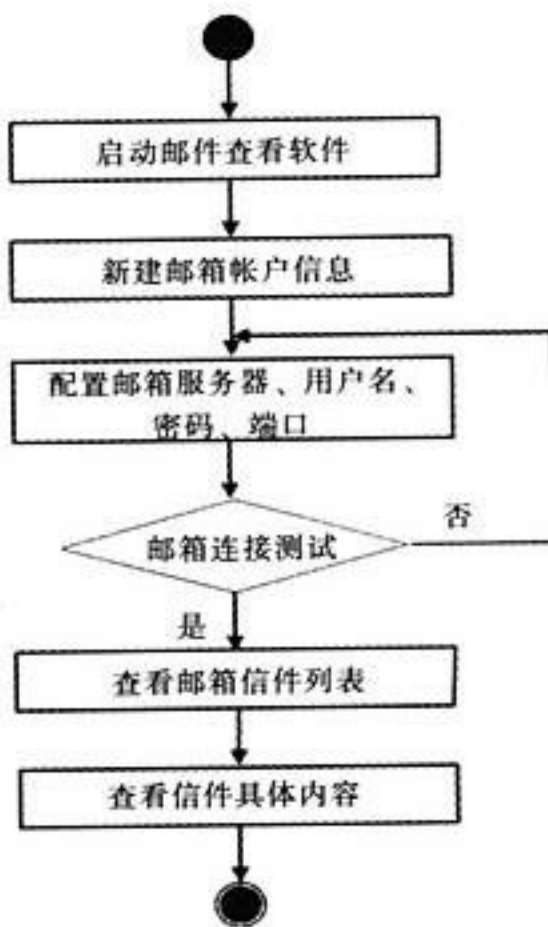


图 2 邮件处理流程图

### 4 功能模块

在邮件查看软件设计中需要用到两个与邮件相关的控件分别是 Psock 和 NMpop3, 在这两个控件当中封装了所有与查阅邮件有关的功能, 如 socket 连接功能、握手协议以及 Windows socket handle API 函数等。在本软件设计中, 只需要声明该类的对象, 调用该类的过程、函数, 或设置对象的一些属性, 便可实现与邮件接收、查看等相关的功能。本软件以 Delphi 为编程工具对邮件查看软件进行设计, 该编程工具最突出的特点是各类控件异常丰富, 拥有强大的控件库, 还具有简单易学、开发软件速度快等特点, 而且构造出的界面漂亮。

根据 Delphi 功能模块实现的依据，邮件查看软件首先创建一个公共模块，这个公共模块封装了该软件所有的公共变量、公共过程和公共函数等，这很好体现了 Delphi 在代码重用方面的功能，本软件采用 XML 格式将邮箱帐户存储到 XML 文件中，启动从 XML 文件中加载。

公共模块程序代码如下：

```
MailRec = Record
    aHost: string[30]; // 邮件服务器
    aUser: string[20]; // 用户名
    aPassWord: string[20]; // 密码
    aPort: string[10]; // 邮箱端口
    SizeArr: Array of string; // 邮箱帐户数组
end;
TPNode = ^MailRec;
MailInfo = Record
    aFrom: string[100]; // 邮件来源
    aSubject: string[100]; // 邮件主题
    MessageID: string[100]; // 邮件 ID
    aHead: string; // 邮件标题
    aBody: string; // 邮件内容
    aDate: string[30]; // 邮件日期
    aSize: string[10]; // 邮件大小
end;
TMainInfo = ^MailInfo;
PPP = Record
    sName: string; // 邮箱帐户名
    tt: Array of MailInfo; // 邮件个数数组
end;
var
    aHost, aUser, aPassWord, aPort: string; // 声明公共变量
// 邮箱服务器、用户名、密码、端口
    aFrom, aSubject, MessageID, aDate, aSize: string; // 声明
// 公共变量邮件来源、邮件主题、邮件 ID、邮件日期、邮件大小
    aHead, aBody: TStrings; // 声明公共变量邮件标题和邮件内容
    aSizeArr: Array of string; // 邮箱帐户数组变量
    Flag: String; // 邮箱连接是否成功
    AMem: Array[1..20] of PPP; // 邮箱帐户记录数组变量
// 获得应用程序的路径
function GetCurrentPath: string;
begin
    Result := ExtractFilePath(Application.ExeName);
end;
// 加载邮箱帐户根目录和子目录信息
procedure InitXML(aPara: string; TreeDir: TTreeView);
var
    i: Integer;
    aNode, tempNode: IXMLNode;
    Node1, Node2: TTreeNode;
    P: TPNode;
    Rec: Integer;
    XMLDoc1: TXMLDocument; // 声明一个 XMLDocument
```

```
// 类的对象变量
begin
    XMLDoc1 := TXMLDocument.Create(nil);
    if aPara = 'Load' then
    begin
        TreeDir.Items.Clear;
        if not FileExists (GetCurrentPath + 'MailPara.XML')
        then // 判断该文件是否存在
        begin
            Node1 := TreeDir.Items.Add(nil, '邮件目录');
            Node1.SelectedIndex := 10;
            Node1.ImageIndex := 10;
            Exit;
        end;
        XMLDoc1.Active := false;
        XMLDoc1.LoadFromFile(getCurrentPath + 'MailPara.XML');
        New(P);
        Node1 := TreeDir.Items.Add(nil, '邮件目录');
        Node1.ImageIndex := 10;
        Node1.SelectedIndex := 10;
        XMLDoc1.Active := true;
        tempNode := XMLDoc1.ChildNodes.FindNode('MailDef');
        if tempNode.ChildNodes.Count = 0 then exit;
        tempNode := tempNode.ChildNodes.Nodes[0];
        // 从 MailPara.XML 文件中读取邮箱帐户信息显示当左边
        // 的树形菜单中
        for i := 0 to tempNode.ChildNodes.Count - 1 do
        begin
            New(P);
            P^.aHost := tempNode.ChildNodes[i].Attributes['HOST'];
            P^.aUser := tempNode.ChildNodes[i].Attributes['USERNAME'];
            P^.aPassWord := tempNode.ChildNodes[i].Attributes
            ['PASSWORD'];
            P^.aPort := tempNode.ChildNodes[i].Attributes['PORT'];
            Node2 := TreeDir.Items.AddChild(Node1, P^.aHost);
            Node2.ImageIndex := 1;
            Node2.SelectedIndex := 2;
            Node2.Data := P;
        end;
        Node1.Expanded := true;
    end;
    // 将树形目录中的邮箱帐户保存为 MailPara.XML 文件
    if aPara = 'Save' then
    begin
        if not FileExists(GetCurrentPath + 'MailPara.XML') then
        begin
            XmlDoc1.XML.Clear;
            XmlDoc1.XML.Add('<?xml version="1.0" encoding="UTF-8"?>');
            XmlDoc1.XML.Add('<MailDef></MailDef>');
            XmlDoc1.Active := true;
            XmlDoc1.SaveToFile(getCurrentPath + 'MailPara.XML');
        end;
```





## NETWORK & COMMUNICATION

```

XMLDoc1.Active := False;
XMLDoc1.LoadFromFile(GetCurrentPath + 'MailPara.XML');
XMLDoc1.Active := true;
aNode := xmlDoc1.ChildNodes.FindNode('MailDef');
aNode.ChildNodes.Clear;
with aNode.addChild('MailDir') do
begin
  Attributes['HOST'] := TreeDir.TopItem.Text;
end;
Node1 := TreeDir.TopItem;
aNode := aNode.ChildNodes[0];
Node1 := Node1.getFirstChild;
Node2 := Node1.GetLastChild;
while Node1 <> Node2 do
begin
  with aNode.AddChild(Node1.Text) do
  begin
    Attributes['HOST'] := TPNODE(Node1.Data).aHost;
    Attributes['USERNAME'] := TPNODE(Node1.Data).aUser;
    Attributes['PASSWORD'] := TPNODE(Node1.Data).aPassWord;
    Attributes['PORT'] := TPNODE(Node1.Data).aPort;
  end;
  Node1 := Node1.getNextSibling;
end;
XMLDoc1.SaveToFile(GetCurrentPath + 'MailPara.XML');
XMLDoc1.Active := false;
end;
end;

```

## 5 功能及代码

### 5.1 软件界面

软件主界面和帐户管理界面如图 3 和图 4 所示。

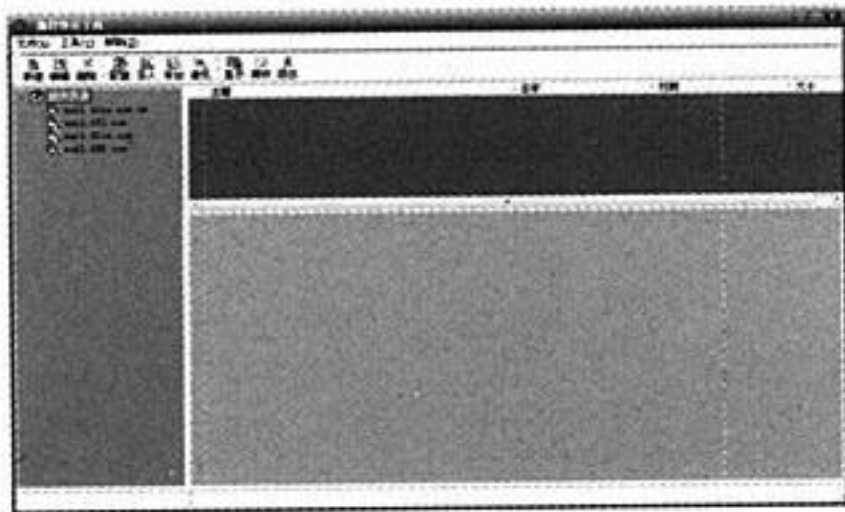


图 3 软件主界面图



图 4 邮箱帐户新建界面

### 5.2 XML 文件格式界面

```

<?xml version="1.0" encoding="UTF-8" ?>
-<mailDef>
-<mailDir HOST="邮件目录">
    <mail.sina.com.cn HOST="mail.sina.com.cn"
    USERNAME="Test1" PASSWORD="Test123" PORT="110" />
    <mail.163.com HOST="mail.163.com" USERNAME="Test1"
    PASSWORD="Test123" PORT="110" />
    <mail.21cn.com HOST="mail.21cn.com" USERNAME="
    Test1" PASSWORD="Test123" PORT="110" />
    <mail.126.com HOST="mail.126.com" USERNAME="Test1"
    PASSWORD="Test123" PORT="110" />
  </mailDir>
</mailDef>

```

### 5.3 窗体程序代码

```

//关闭主应用程序
procedure Tfrm_Main.btnExitClick(Sender: TObject);
begin
  Close;
end;
// 初始化邮箱帐户目录
procedure Tfrm_Main.Init;
var
  aNode: TTreeNode;
begin
  InitXML1('Load');
end;
//在主应用程序关闭前将树形菜单中的邮箱帐户保存为 XML 文件
procedure Tfrm_Main.FormClose (Sender: TObject; var
  Action: TCloseAction);
begin
  aHead.Free;
  aBody.Free;
  InitXML1('Save');
end;
//在主窗体创建的同时创建 TStringList 类对象
procedure Tfrm_Main.FormCreate(Sender: TObject);
begin
  aHead := TStringList.Create;
  aBody := TStringList.Create;
  KFlag := 0;AFlag := 0;
end;
//在主窗体显示的时候加载邮箱帐户信息,同时以进度条的形式
//显示加载的进度
procedure Tfrm_Main.FormShow(Sender: TObject);
begin
  Init;
  showProgress(pBar,5,30,true);
end;
// 实现新建一个邮箱帐户的功能
procedure Tfrm_Main.btnNewClick(Sender: TObject);

```



```

var
  aNode: TTreeNode;
  P: TPNODE; // 声明一个邮箱帐户指针记录邮箱帐户信息
begin
  New(P);
  flag := 'new';
  Try
    frm_Account := Tfrm_Account.Create(nil); // 创建邮箱帐户
//管理窗口
    if frm_Account.ShowModal = mrOK then
      begin
        aNode := Tree.Items.AddChild(Tree.TopItem, aHost);
        aNode.ImageIndex := 1;
        aNode.SelectedIndex := 2;
        P^.aHost := aHost;
        P^.aUser := aUser;
        P^.aPassWord := aPassWord;
        P^.aPort := aPort;
        aNode.Data := P;
      end;
    finally
      InitXML1('Save'); // 录入邮箱帐户信息完毕,保存为 XML 文件
    end;
end;
//获取邮件列表信息
procedure Tfrm_Main.GetMailInfo(iRec: integer);
var
  i: integer;
function GetDate(aStr: TStrings): string; // 获取邮件日期
var
  a: integer;
begin
  for a := 0 to aStr.Count - 1 do
    if Pos('Date:', aStr.Strings[a]) > 0 then
      begin
        Result := Copy(aStr.Strings[a], 6, 26);
      end;
  end;
end;
begin
  DM.P_Accept.GetMailMessage(iRec);
  aFrom := DM.P_Accept.MailMessage.From; //获取邮件来源
  aSubject := DM.P_Accept.MailMessage.Subject; //获取邮
//件主题
  MessageID := DM.P_Accept.MailMessage.MessageID;
//获取邮件 ID
  aHead.Assign(DM.P_Accept.MailMessage.Head); // 获取邮
//件头
  aBody.Assign(DM.P_Accept.MailMessage.Body); // 获取邮
//件内容
  aDate := GetDate(aHead); // 获取邮件日期
  aSize := aSizeArr[iRec]; //获取邮件大小
end;

```

```

// 获取邮件附件信息
procedure Tfrm_Main.GetSummInfo(iRec: integer);
begin
  DM.P_Accept.GetSummary(iRec);
  aFrom := DM.P_Accept.Summary.From;
  aSubject := DM.P_Accept.Summary.Subject;
  aSize := IntToStr(DM.P_Accept.Summary.Bytes);
  MessageID := DM.P_Accept.Summary.MessageID;
  aHead.Assign(DM.P_Accept.Summary.Header);
end;
// 邮箱帐户连接
procedure Tfrm_Main.ConnectIt(aNode: TTreeNode);
begin
  DM.P_Accept.AttachFilePath := '';
  DM.P_Accept.DeleteOnRead := FALSE;
  DM.P_Accept.TimeOut := 20000;
  DM.P_Accept.Host := TPNODE(aNode.Data)^.aHost;
  DM.P_Accept.Port := StrToInt(TPNODE(aNode.Data)^.aPort);
  DM.P_Accept.UserID := TPNODE(aNode.Data)^.aUser;
  DM.P_Accept.PassWord := TPNODE(aNode.Data)^.aPassWord;
  Try
    DM.P_Accept.Connect;
  Except
    On E: Exception do
      begin
        ShowMessage(E.Message);
        Exit;
      end;
  end;
end;
// 在邮箱帐户列表中选中邮箱帐户,在右边显示邮件列表
procedure Tfrm_Main.MailAccountTreeClick(Sender: TObject);
var
  aNode: TTreeNode;
  aItem: TListItem;
  i: integer;
  P: TMainInfo;
begin
  aNode := Tree.Selected;
  if aNode = nil then exit;
  if aNode.Level = 0 then exit;
  DM.P_Accept.Disconnect;
  StatusBar1.Panels[0].Text := '正在连接...';
  Application.ProcessMessages;
  ConnectIt(aNode);
  if DM.P_Accept.Connected then
    begin
      ListMail.Items.Clear;
      StatusBar1.Panels[0].Text := '准备开始加载信息...';
      Application.ProcessMessages;
      for i := 1 to DM.P_Accept.MailCount - 1 do
        begin

```





## NETWORK &amp; COMMUNICATION

```

New(P);
GetSummInfo(i);
altem := ListMail.Items.Add;
altem.Caption := IntToStr(i+1);
altem.SubItems.Add(aFrom);
altem.SubItems.Add(aSubject);
altem.SubItems.Add(aDate);
altem.SubItems.Add(aSize);
P^.aFrom := aFrom;
P^.aSubject := aSubject;
P^.aHead := aHead.Text;
P^.aBody := aBody.Text;
P^.aDate := aDate;
P^.aSize := aSize;
altem.Data := p;
end;
end;
end;
// 选中某个邮件,则在 DocView 控件中显示该邮件的具体内容
procedure Tfrm_Main.ListMailSelectItem (Sender: TObject;
Item: TListItem;
Selected: Boolean);
begin
if Item = nil then exit; //如果未选中则退出
DocView.Lines.Text := TMainInfo(Item.Data)^.aBody;
end;
// 实现显示邮箱帐户的进度
procedure Tfrm_Main.ShowProgress (ABar: TProgressBar;
iPos, aLen: Integer;
aFlag: Boolean);
begin
ABar.Visible := aFlag;
ABar.Parent := StatusBar1;
ABar.Position := iPos;
ABar.Max := aLen;
ABar.Left := StatusBar1.Left;
ABar.Top := StatusBar1.Top;
ABar.Width := 200;
ABar.Height := StatusBar1.Height;
end;
// 实现删除邮箱帐户的功能
procedure Tfrm_Main.btnDelClick(Sender: TObject);
var
aNode: TTreeNode;
DelStr: string;
begin
aNode := Tree.Selected;
if (aNode = nil) or (aNode.Level=0) then exit;
DelStr := '你确认要删除['+aNode.Text+']吗? ';
if Application.MessageBox (PChar (DelStr), '删除确认 ',
MB_OKCANCEL) = IDCANCEL then exit;
if FindIt(aNode.Text) then

```

```

begin
AMem[AFlag].sName := '';
AMem[AFlag].tt := nil;
end;
aNode.Delete;
end;
// 实现修改邮箱帐户的功能
procedure Tfrm_Main.btnEditClick(Sender: TObject);
var
P: TPNODE;
aNode: TTreeNode;
begin
Flag := 'edit';
aNode := Tree.Selected;
aHost := TPNODE(aNode.Data)^.aHost;
aUser := TPNODE(aNode.Data)^.aUser;
aPassWord := TPNODE(aNode.Data)^.aPassWord;
aPort := TPNODE(aNode.Data)^.aPort;
if (aNode = nil) or (aNode.Level=0) then exit;
Try
frm_Account := Tfrm_Account.Create(nil);
if frm_Account.ShowModal = mrOK then
begin
new(P);
aNode.Text := aHost;
P^.aHost := aHost;
P^.aUser := aUser;
P^.aPassWord := aPassWord;
P^.aPort := aPort;
aNode.Data := P;
end;
finally
InitXML1('Save');
end;
end;
//实现从 XML 文件导入 邮箱帐户信息
procedure Tfrm_Main.btnImportClick(Sender: TObject);
var
aNode, tempNode: IXMLNode;
Node1, Node2: TTreeNode;
P: TPNODE;
Rec,i: Integer;
begin
if OpenFileDialog1.Execute then
if Trim(OpenDialog1.FileName) <> '' then
begin
Tree.Items.Clear;
if not FileExists(OpenDialog1.FileName) then // 新建的
//邮件存放在 Mail.XML 文件中
begin
Node1 := Tree.Items.Add(nil, '邮件目录 ');
Node1.SelectedIndex := 10;

```



```

Node1.ImageIndex := 10;
Exit;
end;
XMLDoc1.Active := false;
XMLDoc1.LoadFromFile(OpenDialog1.FileName); //从邮件
//目录 XML 文件获取
New(P);
Node1 := Tree.Items.Add(nil, '邮件目录');
Node1.ImageIndex := 10;
Node1.SelectedIndex := 10;
XMLDoc1.Active := true;
tempNode := XMLDoc1.ChildNodes.FindNode('MailDef');
if tempNode.ChildNodes.Count = 0 then exit;
tempNode := tempNode.ChildNodes.Nodes[0];
// 加载邮件子目录信息
for i := 0 to tempNode.ChildNodes.Count - 1 do
begin
    New(P);
    P^.aHost := tempNode.ChildNodes[i].Attributes['HOST'];
    P^.aUser := tempNode.ChildNodes[i].Attributes
['USERNAME'];
    P^.aPassWord := tempNode.ChildNodes[i].Attributes
['PASSWORD'];
    P^.aPort := tempNode.ChildNodes[i].Attributes['PORT'];
    Node2 := Tree.Items.AddChild(Node1, P^.aHost);
    Node2.ImageIndex := 1;
    Node2.SelectedIndex := 2;
    Node2.Data := P;
end;
Node1.Expanded := true;
end;
end;
// 根据邮箱帐户名称查找邮箱
function Tfrm_Main.FindIt(aStr: string): Boolean;
var
    i: integer;
begin
    Result := false;
    for i := 1 to 19 do
    if AMem[i].sName = aStr then
    begin
        AFlag := i;
        Result := true;
        exit;
    end;
    end;
end;
// 查看某个邮箱帐户的邮件列表信息
procedure Tfrm_Main.btnShowClick(Sender: TObject);
var
    aNode: TTreeNode;
    altem: TListItem;
    i: integer;

```

```

P: TMainInfo;
begin
    aNode := Tree.Selected;
    if aNode = nil then exit;
    if aNode.Level = 0 then exit;
    if FindIt(aNode.Text) then
    begin
        for i := 1 to High(AMem[AFlag].tt) do
        begin
            altem := ListMail.Items.Add;
            altem.Caption := IntToStr(i+1);
            altem.SubItems.Add(aSubject);
            altem.SubItems.Add(aFrom);
            altem.SubItems.Add(aDate);
            altem.SubItems.Add(aSize);
            P^.aFrom := aFrom;
            P^.aSubject := aSubject;
            P^.aHead := aHead.Text;
            P^.aBody := aBody.Text;
            P^.aDate := aDate;
            P^.aSize := aSize;
            altem.Data := p;
        end;
        exit;
    end;
    StatusBar1.Panels[0].Text := '正在连接,稍等...';
    Application.ProcessMessages;
    DM.P_Accept.DisConnect;
    ConnectIt(aNode);
    StatusBar1.Panels[0].Text := '连接成功。';
    Application.ProcessMessages;
    if not DM.P_Accept.Connected then
        StatusBar1.Panels[0].Text := '连接失败。';
    StatusBar1.Panels[0].Text := '正在获取每个邮件大小。';
    Application.ProcessMessages;
    DM.P_Accept.List;
    if DM.P_Accept.Connected then
    begin
        ListMail.Items.Clear;
        StatusBar1.Panels[0].Text := '开始加载邮件列表信息。';
        Application.ProcessMessages;
        AMem[KFlag].sName := aNode.Text;
        SetLength(AMem[KFlag].tt, DM.P_Accept.MailCount+1);
        for i := 1 to DM.P_Accept.MailCount - 1 do
        begin
            New(P);
            GetMailInfo(i);
            StatusBar1.Panels[0].Text := '稍后,正在加载信息:' +
            IntToStr(i)+'/'+IntToStr(DM.P_Accept.MailCount);
            Application.ProcessMessages;
            ShowProgress(pBar,i,DM.P_Accept.MailCount,True);
            altem := ListMail.Items.Add;

```





## NETWORK &amp; COMMUNICATION

```

altem.Caption := IntToStr(i+1);
altem.SubItems.Add(aSubject);
altem.SubItems.Add(aFrom);
altem.SubItems.Add(aDate);
altem.SubItems.Add(aSize);
P^.aFrom := aFrom;
P^.aSubject := aSubject;
P^.aHead := aHead.Text;
P^.aBody := aBody.Text;
P^.aDate := aDate;
P^.aSize := aSize;
altem.Data := p;
AMem[KFlag].tt[i].aFrom := aFrom;
AMem[KFlag].tt[i].aSubject := aSubject;
AMem[KFlag].tt[i].aHead := aHead.Text;
AMem[KFlag].tt[i].aBody := aBody.Text;
AMem[KFlag].tt[i].MessageID := MessageID;
AMem[KFlag].tt[i].aDate := aDate;
AMem[KFlag].tt[i].aSize := aSize;
end;
end;
KFlag := KFlag + 1;
end;
// 实现邮箱帐户连接测试功能
procedure Tfrm_Account.btnConnectClick(Sender: TObject);
begin
  DM.P_Accept.AttachFilePath := '';
  DM.P_Accept.DeleteOnRead := FALSE;
  DM.P_Accept.TimeOut := 20000;
  DM.P_Accept.Host := Trim(edtHost.Text);
  DM.P_Accept.Port := StrToInt(Trim(edtPort.Text));
  DM.P_Accept.UserID := Trim(edtUserName.Text);
  DM.P_Accept.Password := Trim(edtPassWord.Text);
  try
    DM.P_Accept.Connect;
  except
    On E: Exception do
      begin
        showMessage(E.Message);
        exit;
      end;
    end;
  end;
  Label1.caption := '# of Messages: '+IntToStr(DM.P_Accept.
MailCount);
  if DM.P_Accept.Connected then
    showMessage('连接成功')
  else
    showMessage('连接失败');
  end;
// 在该窗口关闭前,保存邮箱地址,用户名,密码,端口信息等
procedure Tfrm_Account.FormClose(Sender: TObject;
var Action: TCloseAction);

```

```

begin
  aUser := Trim(edtUserName.Text);
  aPassWord := Trim(edtPassWord.Text);
  aHost := Trim(edtHost.Text);
  aPort := Trim(edtPort.Text);
end;
//只有在邮箱地址,用户名,密码,端口信息都输入内容确定按
//钮和连接测试按钮才能使用
procedure Tfrm_Account.edtHostChange(Sender: TObject);
begin
  btnOK.Enabled := (Trim (edtHost.Text) <> '') and (Trim
(edtUserName.Text) <> '') and (Trim (edtPassWord.Text) <> '')
and(Trim(edtPort.Text)<> '');
  btnConnect.Enabled := (Trim (edtHost.Text) <> '') and (Trim
(edtUserName.Text) <> '') and (Trim (edtPassWord.Text) <> '')
and(Trim(edtPort.Text)<> '');
end;
procedure Tfrm_Account.FormCreate(Sender: TObject);
begin
//新建一个邮箱帐户时,清空邮箱地址,用户名,密码,端口信息
if flag = 'new' then
  begin
    edtHost.Text := '';
    edtUserName.Text := '';
    edtPassWord.Text := '';
    edtPort.Text := '110';
  end;
//修改一个邮箱帐户时,将要修改的邮箱地址,用户名,密码,
//端口信息 加载到编辑控件中
if flag = 'edit' then
  begin
    edtHost.Text := aHost;
    edtUserName.Text := aUser;
    edtPassWord.Text := aPassWord;
    edtPort.Text := aPort;
  end;
end;
end;

```

## 6 结语

借助于 Psock 控件和 NMPop3 控件强大的功能实现了软件的新建邮箱帐户、邮箱帐户连接测试、查阅邮件内容及邮件附件内容等功能。体现了 Delphi 编程风格、语法特点等。在企业项目实施中经常用到的代码重用以及命名规范等也有了很好的体现。

## 参考文献

- [1] 牛汗民. Delphi7 应用开发教程. 北京: 科学出版社, 2005.  
(收稿日期: 2012-08-09)



## DirectShowLib 的家庭视频监控系统

陶 阳

**摘 要:** DirectShowLib 是对微软 DirectShow 的 C# 开源封装。基于 DirectShowLib 类库, 设计并实现了一个家庭视频监控系统原型, 通过对 USB 无驱摄像头的操控, 实现视频的采集、抓图和录像功能, 可满足一般家庭安防系统中的实时视频监控的需要。

**关键词:** DirectShow 开发包; 视频监控; 家庭安防; C# 语言

### 1 引言

视频监控是通过获取监控目标的视频图像信息, 并对其进行监视、记录、回溯, 并根据视频图像信息人工或自动地做出相应的动作, 以达到对监控目标的监视、控制、安全防范和智能管理等目的。随着网络通信技术、图像压缩处理技术以及传输技术的快速发展, 使得家庭能够采用最新的视频压缩和图像处理技术, 通过 USB 摄像头采集视频图像, 可实现家庭环境的实时视频监控。

为了免除越来越多的城市白领阶层对家人健康和家庭财产安全的后顾之忧, 市场上有不少家庭视频监控系统, 比如基于数字视频录像机 DVR 的网络化家庭监控系统、环球眼视频监控系统等。虽然它可使得你无论身在全球何处都可以通过互联网和全球移动通信系统随时监控家庭的安全环境, 但是这样的监控系统部署困难, 价格昂贵, 对一般家庭的计算机用户来说, 不太适合。一般用户来说, 家庭拥有普通计算机、网络 and USB 摄像头, 而市场上的视频监控系统一般要求的是专业的摄像头和网络终端, 而且视频监控软件的价格比较高, 对普通用户来说, 必要性不大。

因此, 针对一般家庭用户, 一种简单有效的家庭视频监控系统更具实际意义, 可方便用户自行设计、搭建一个比较实用的家庭安防系统。DirectShowLib 是一个对微软的功能强大的 DirectShow 的 C# 开源封装的类库, 可实现对视频音频的采集、编码、压缩和处理。一般家庭用户中, 使用的摄像头基本上是 USB 的无驱摄像头, 所以可以基于该类库, 在 VS2008 开发环境下, 使用 C# 语言, 操作控制 USB 摄像头, 设计并实现一套家庭视频监控系统, 通过视频采集、录像等功能, 实现对家庭环境的视频监控。

### 2 DirectShowLib 类库

DirectShowLib 是 kawaic, snarfle 等人对微软 DirectShow 的 C# 开源封装, 目的是使 C# 开发者也能方便地使用 DirectShow。有了该 lib, 在 C# 下开发视频采集等已经不再是难事。DirectShow

是微软公司在 ActiveMovie 和 Video for Windows 的基础上推出的新一代基于 COM (Component Object Model) 的流媒体处理的开发包, 与 DirectX 开发包一起发布。DirectShow 使用一种叫 Filter Graph 的模型来管理整个数据流的处理过程, 运用 DirectShow, 可以很方便地从支持 WDM 驱动模型的采集卡上捕获数据, 并且进行相应的后期处理乃至存储到文件中。它广泛地支持各种媒体格式, 包括 Asf、Mpeg、Avi、Dv、Mp3、Wave 等, 为多媒体流的捕捉和回放提供了强有力的支持。

DirectShow 运行的方式通常是一个开发者创建一个 Filter Graph, 把一些 Filter 加入 Filter Graph, 然后播放文件, 或者播放来自互联网或照相机的数据。当播放进程运行时, Filter Graph 在 Windows 注册中寻找注册了的 Filters, 并且为这些 Filter 创建本地提供的 Graph。在这之后, 它将所有的 Filter 连接在一起, 并且在开发者的请求下, 播放/中止创造的 Graph。在 DirectShow 中有许多抽象的播放源文件的方法, 实现这些功能也是相当简单的而且不需要一个定制过的 Filter。

既然 DirectShowLib 是对 DirectShow 的封装, 那么它自然也可以实现 DirectShow 的各种功能, 其使用和调用方法与 DirectShow 是一样的。这里仅介绍在视频监控系统中必要的几种接口对象的方法, 具体实现中, 这些方法都包含在一个单独的类文件中, 如 Capture.cs。

(1) 获取视频采集设备 IBaseFilter 接口对象:

```
//获取所有视频设备名称
public ArrayList GetVideoInputDevice()
{
    return GetDeviceCollection(FilterCategory.VideoInputDevice);
}

private ArrayList GetDeviceCollection(Guid DeviceType)
{
    ArrayList returnString = new ArrayList();
    foreach (DsDevice ds in DsDevice.
        GetDevicesOfCat(DeviceType))
    {
        returnString.Add(ds.Name);
    }
}
```





## GRAPHICS AND IMAGE PROCESSING

```

    }
    return returnString;
}
//通过获取到的视频设备名称设置采集设备的 IBaseFilter 对象
public bool SetVideoInputDevice(string VideoInput
DeviceName)
{
    //创建视频输入设备接口
    theVideoDevice = CreateFilter (FilterCategory.
VideoInputDevice, VideoInputDeviceName);
}
//通过过滤器类型和过滤器名称获取 IBaseFilter 接口
private IBaseFilter CreateFilter (Guid category, string
friendlyname)
{
    object source = null;
    Guid iid = typeof(IBaseFilter).GUID;
    foreach (DsDevice device in DsDevice.
GetDevicesOfCat(category))
    {
        if (device.Name.CompareTo(friendlyname) == 0)
        {
            device.Mon.BindToObject(null, null, ref iid, out source);
            break;
        }
    }
    return (IBaseFilter)source;
}
(2) 初始化接口对象:
private void InitInterfaces()
{
    int hr = 0;
    // 获取 IGraphBuilder 接口对象
    this.m_graphBuilder = (IGraphBuilder)new FilterGraph();
    //获取 ICaptureGraphBuilder2 接口对象
    this.m_captureGraphBuilder= (ICaptureGraph
Builder2) new CaptureGraphBuilder2();
    //获取 m_graphBuilder 接口对象的 IMediaControl 对象
    this.m_mediaControl = (IMediaControl)this.m_graphBuilder;
    //获取 m_graphBuilder 接口对象的 IVideoWindow 对象
    this.m_videoWindow = (IVideoWindow)this.m_graphBuilder;
    //获取 m_graphBuilder 接口对象的 IMediaEventEx 对象
    this.m_mediaEventEx = (IMediaEventEx)this.m_graphBuilder;
    //设置 ICaptureGraphBuilder2 的 IGraphBuilder 接
    //口对象为当前对象
    hr = this.m_captureGraphBuilder.SetFiltergraph
(this.m_graphBuilder);
    DsError.ThrowExceptionForHR(hr);
    //注册事件到应用程序的窗体上
    hr=this.m_mediaEventEx.SetNotifyWindow (this.
hwnPropertyPageOwner, WM_GRAPHNOTIFY, IntPtr.Zero);
    DsError.ThrowExceptionForHR(hr);
}

```

```

    }
(3) 视频预览:
public void VideoPreview()
{
    try
    {
        int hr = 0;
        hr = this.m_graphBuilder.AddFilter
(theVideoDevice, "Video Capture");
        DsError.ThrowExceptionForHR(hr);
        // 通过 theVideoDevice (IBaseFilter) 视频接口对象
        //的 Preview Pin 预览
        hr = this.m_captureGraphBuilder.RenderStream (PinCategory.
Preview, MediaType.Video, theVideoDevice, null, null);
        DsError.ThrowExceptionForHR(hr);
        //插入 SampleGrabber
        m_pinStill = DsFindPin.ByCategory
(theVideoDevice, PinCategory.Still, 0);
        if (m_pinStill == null)
        {
            m_pinStill = DsFindPin.ByCategory
(theVideoDevice, PinCategory.Capture, 0);
        }
        // 获取 theVideoDevice 的 IAMVideoControl 对象,
        //对于具有 Still Pin 的对象可以获到,采集设备不具备 Still Pin,
        //那么该对象将为 Null
        m_VidControl = theVideoDevice as IAMVideoControl;
        // 设置采集视频参数
        if (this.videoHeight + this.videoWidth + this.videoStride > 0)
        {
            SetConfigParms(m_pinStill, this.videoWidth, this.videoHeight,
24);
        }
        //开始拍照功能所需的接口对象
        // 获得 SampleGrabber 对象接口
        sampGrabber = new SampleGrabber() as ISampleGrabber;
        // 配置 sample grabber
        baseGrabFlt = sampGrabber as IBaseFilter;
        ConfigureSampleGrabber(sampGrabber);
        // 将 sample grabber 添加到图形过滤器中
        hr = m_graphBuilder.AddFilter(baseGrabFlt, "Ds.NET Grabber");
        DsError.ThrowExceptionForHR(hr);
        //通过渲染将采集设备的相关输出 Pin 与 sample
        //grabber 对象的输入 Pin 连接起来
        //如果采集设备提供 Still Pin,则通过 Still Pin 连接,
        //否则直接使用 Capture Pin 连接
        if (m_VidControl != null)
        {
            hr = this.m_captureGraphBuilder.RenderStream (PinCategory.
Still, MediaType.Video, theVideoDevice, null, baseGrabFlt);
            DsError.ThrowExceptionForHR(hr);
        }
    }
}

```



```

else
{
    hr = this.m_captureGraphBuilder.RenderStream
(PinCategory.Capture, MediaType.Video, theVideoDevice,
null, baseGrabFlt);
    DsError.ThrowExceptionForHR(hr);
}
//设置抓取图片相关参数
SaveSizeInfo(sampGrabber);
//拍照功能所需的接口对象添加结束
//开始将视频窗口绑定到主窗体上
hr = this.m_videoWindow.put_Owner (this.
hwnVideoWindowOwner);
DsError.ThrowExceptionForHR(hr);
hr = this.m_videoWindow.put_WindowStyle
(WindowStyle.Child | WindowStyle.ClipChildren);
DsError.ThrowExceptionForHR(hr);
if (this.m_videoWindow != null)
{
    this.m_videoWindow.SetWindowPosition(0, 0,
this.videoWidth, this.videoHeight);
}
hr = this.m_videoWindow.put_Visible(OABool.True);
DsError.ThrowExceptionForHR(hr);
//开始预览采集设备采集到的视频
hr = this.m_mediaControl.Run();
DsError.ThrowExceptionForHR(hr);
m_IsPreview = true;
}
catch
{
    m_IsPreview = false;
    throw new Exception ("VideoPreview 函数出现异
常,视频预览失败!");
}
}

```

(4) 抓图:

```

public Bitmap SnapeBitmap()
{
    if (this.m_IsPreview)
    {
        if (this.m_ipBuffer != IntPtr.Zero)
        {
            Marshal.FreeCoTaskMem(this.m_ipBuffer);
            this.m_ipBuffer = IntPtr.Zero;
        }
        this.m_PictureReady.Reset();
        //分配空间存储抓取的图片
        this.m_ipBuffer = Marshal.AllocCoTaskMem
(Math.Abs(this.videoStride) * this.videoHeight);
        try
        {

```

```

this.m_WantOne = true;
if (this.m_VidControl != null)
{
    //启动图片抓拍
    DsError.ThrowExceptionForHR (this.m_VidControl.
SetMode(this.m_pinStill, VideoControlFlags.Trigger));
}
if (! this.m_PictureReady.WaitOne(0xbb8, false))
{
    throw new Exception("获取图片超时");
}
}
catch
{
    Marshal.FreeCoTaskMem(this.m_ipBuffer);
    this.m_ipBuffer = IntPtr.Zero;
    //throw;
}
if (this.m_ipBuffer != IntPtr.Zero)
{
    Bitmap bitmap = new Bitmap(this.videoWidth,
this.videoHeight, this.videoStride, PixelFormat.Format24bpp
Rgb, this.m_ipBuffer); //图片转存到 Bitmap 对象
    bitmap.RotateFlip(RotateFlipType.Rotate180FlipX); //图片翻转
    return bitmap;
}
}
return null;
}

```

(5) 视频录像:

```

public bool RecordVideo(string mSaveFileName)
{
    if (this.m_IsRecord)
    {
        throw new Exception("正在录制视频,请停止后重新录制!");
    }
    try
    {
        IBaseFilter ppbf;
        IFileSinkFilter ppSink;
        if (! this.m_IsPreview)
        {
            this.VideoPreview();
        }
        this.m_mediaControl.Stop();
        this.m_IsRecord = true;
        if (this.m_VidControl == null)
        {
            //去除 Filter baseGrabFlt
            DsError.ThrowExceptionForHR (this.m_graphBuilder.
RemoveFilter(this.baseGrabFlt));
        }
    }
}

```





## GRAPHICS AND IMAGE PROCESSING

```

//添加音频设备的 Filter
DsError.ThrowExceptionForHR (this.m_graphBuilder.
AddFilter(this.theAudioDevice, "Audio Input Device"));
//添加视频压缩 Filter
DsError.ThrowExceptionForHR (this.m_graphBuilder.AddFilter
(this.theVideoCompressor, "Video compressor filter"));
//添加音频压缩 Filter
DsError.ThrowExceptionForHR (this.m_graphBuilder.AddFilter
(this.theAudioCompressor, "Audio compressor filter"));
//设置视频录像输出文件参数
DsError.ThrowExceptionForHR (this.
m_captureGraphBuilder.SetOutputFileName (MediaSubType.
Avi, mSaveFileName, out ppbf, out ppSink));
//Render 视频设备和压缩 Filter
DsError.ThrowExceptionForHR (this.
m_captureGraphBuilder.RenderStream (PinCategory.Capture,
MediaType.Video, this.theVideoDevice, this.theVideoCompre
ssor, ppbf));
//Render 音频设备和压缩 Filter
DsError.ThrowExceptionForHR (this.m_captureGraphBuilder.
RenderStream (PinCategory.Capture, MediaType.Audio, this.
theAudioDevice, this.theAudioCompre
ssor, ppbf));

this.m_mediaControl.Run();
}
catch (Exception exception)
{
this.StopRecordVideo();
this.m_IsRecord = false;
throw new Exception(exception.Message + "
\nRecordVideo 函数出现异常,视频录制失败!");
}
return true;
}

```

## 3 视频监控系统架构

视频监控系统主要应包括以下功能模块：视频源选择、视频图像采集、监控视频录像、监控视频抓图等。视频图像采集主要依靠连接电脑的摄像头来完成，通过控制视频输入设备的选择，找到可用的视频源，而后根据 DirectShow 的要求，进行连接，即可完成视频源的选择、图像采集以及视频预览等功能。系统设计的框架如图 1 所示。

视频图像的信息量是巨大的。对视频图像进行压缩是必须的。视频采用的压缩标准有：

(1) 静止图像压缩技术：JPEG，M-JPEG。采用小波变换等技术，运用帧内处理技术。静止图像较清晰，但对连续运动图像，文件占用的带宽和硬盘都很大。

(2) 运动图像压缩技术：MPEG-1，MPEG-2，H.263，MPEG-4。

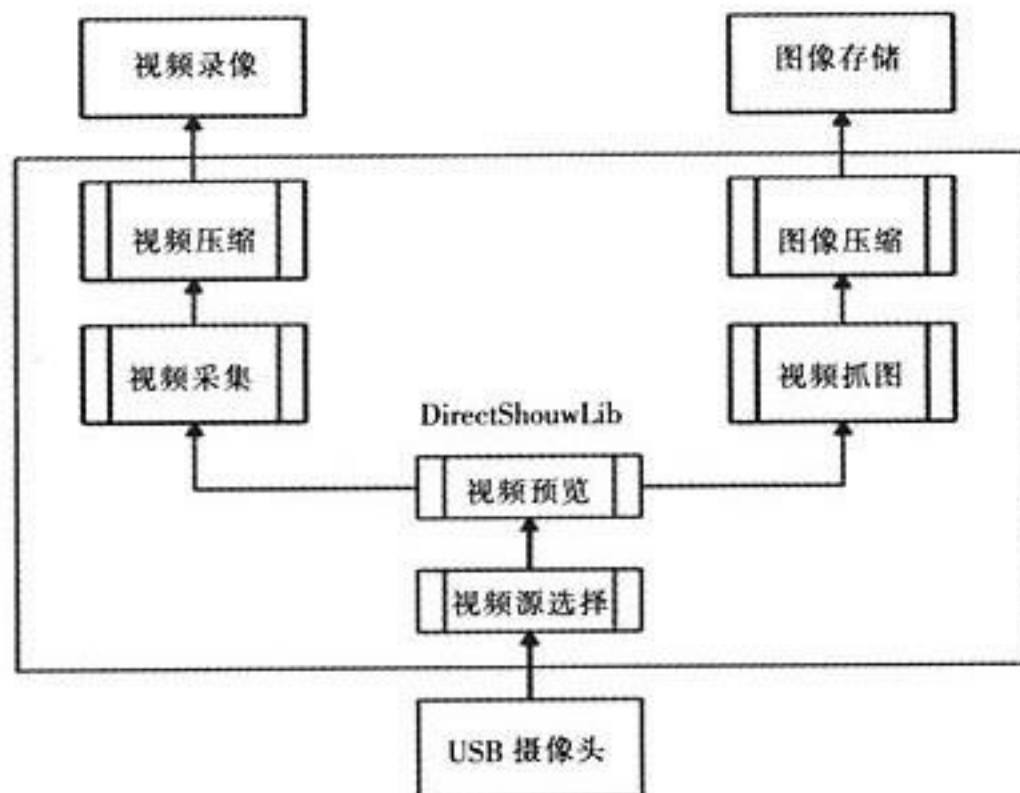


图 1 基于 DirectShowLib 的视频监控系统架构

在 DirectShow 中有现成的已经实现的视频图像的压缩算法，在进行视频图像处理的时候，直接选择相应的可用的压缩算法即可。其中的音频采集也是一样的道理，这里省略。

## 4 视频监控系统程序实现

系统实现是在 VS2008 开发环境中进行的。首先，在 VS2008 中，选择 C# 语言，建立一个解决方案，命名为 Family Capture。然后，加载 DirectShowLib-2008 工程。该工程是从开源网站下载。接着，建立主工程，命名为 FamilyCaptureProject，封装与 DirectShowLib 的接口函数到一个单独的类文件中，如 Capture.cs。整个解决方案在 VS2008 环境下的视图如图 2 所示。



图 2 系统的解决方案物理视图

在系统的主窗体中，可以看到主界面，如图 3 所示。从界面可以看出，系统主要包括抓图和录像两个功能。

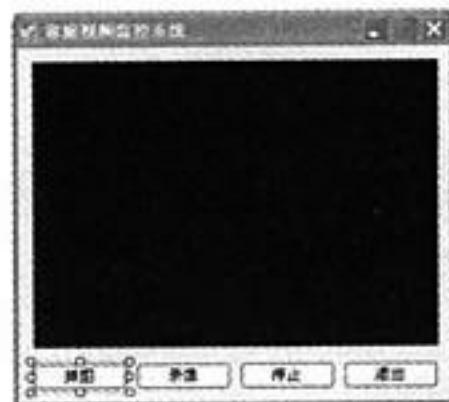


图 3 系统的主界面



抓图功能是系统的一个可扩展的功能,从监控的目的来看,可以设置定时或者根据移动监测算法主动抓图,这样就可以很容易监测到监控区域的移动物体或对象。这里,仅给出抓图的一般算法,如下:

```
Bitmap bitmap = this.m_Capture.SnapeBitmap();
if (bitmap != null)
{
    Image newImage = bitmap.GetThumbnailImage (320, 240,
    new Image.GetThumbnailImageAbort(IsTrue), IntPtr.Zero);
    //创建图像
    Bitmap image = new Bitmap(newImage);
    //获取图像的 BitmapData 对象
    BitmapData data = image.LockBits(new Rectangle(0, 0,
    image.Width,image.Height), ImageLockMode.ReadWrite, Pix
    elFormat.Format24bppRgb);
    this.pictureBox1.Image = newImage;// Image.
    FromHbitmap(bitmap.GetHbitmap());
    image.Save(this.GetFilePath() + ".jpg", ImageFormat.Jpeg);
}
```

录像功能是视频监控系统的最基本的功能。其实现算法如下:

```
try
{
    //选择音频的压缩 Filter
    this.m_Capture.SetAudioCompressor("MPEG Layer-3");
    //选择视频的压缩 Filter
    this.m_Capture.SetVideoCompressor("DivX ;
    -) MPEG-4 Low-Motion");
    //选择视频源设备
    this.m_Capture.SetVideoInputDevice (this.
    cb_VideoInputDevice.SelectedItem.ToString());
    //视频预览
    this.m_Capture.VideoPreview();
    //视频捕获、压缩、输出
    this.m_Capture.RecordVideo(this.GetFilePath() + ".avi");
    this.btnStop.Enabled = true;
    this.btnRecord.Enabled = false;
}
catch (Exception exception)
{
    MessageBox.Show(exception.Message);
}
```

通过程序编译、调试,最终的运行结果如图4所示。系统运行后,自动进入到监控视频的预览状态。在该界面上,点击抓图,会在程序所在目录下输出相应的图像文件,并在窗口侧边显示抓图效果。如图5所示。如果点击录像,系统会进入到录像状态,直至点击停止,系统在程序所在目录下输入所录制的视频文件。当然,在实际应用中,可以设置录制时间,或者设置录像的存储原则等,以更灵活地满足视频

监控的需要。



图4 系统运行主界面

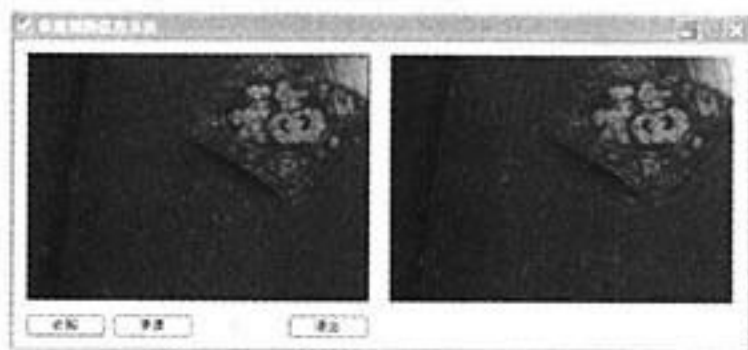


图5 系统抓图后的运行效果图

## 5 结语

DirectShowLib 的功能强大,家庭视频监控系统仅使用了其中的一部分功能。如果要改进系统中的视频压缩算法,更好地满足视频监控的需要,可以在此基础上展开研究,实现更好的视频压缩 Filter,加载到系统中就可以了。

基于 DirectShowLib 开发的这样的家庭视频监控系统,可以满足一般家庭用户的需要。方法简单,通过普通的 USB 摄像头,就可以在家中建立一个视频监控系统。当然,这仅仅是一个原型系统,还有许多功能需要实现,以满足更多用户的需要,比如网络监控、远程监控、移动监测等功能,这就是下一步工作的研究方向。

## 参考文献

- [1] 王宪,刘井权,戴旻.基于 ARM 平台的远程视频监控系统的研究.微计算机信息,2007,23 (2).
- [2] 郭家驹.基于 P2P 原理的网络球形摄像机的关键技术研究[D].浙江大学机械与能源工程学院,浙江大学,2008.
- [3] 段军棋,蒋丹.远程视频监控系统的设计与实现.电子科技大学学报,2002,31 (5).
- [4] 宋玉峰,周泓.远程数字视频监控系统的设计与实现.计算机工程,2002,8.
- [5] 李广伦,爻伟群.视频监控系统中运动目标的实时检测.计算机工程,2009,35 (17).
- [6] 张文杰,戚飞虎,江卓军.实时视频监控系统中运动目标检测和跟踪的一种实用方法.上海交通大学学报,2002,12.

(收稿日期:2012-09-21)





# 用 VB 实现 PNG 四态按钮控件

曹卫忠

**摘要:** 介绍了用 VB 实现 PNG 四态按钮控件的功能, 分析了实现难点, 提供了解决思路和编程方法。

**关键词:** VB 语言; PNG 格式; 按钮控件

## 1 引言

因项目要求按钮能够载入 Png 格式图片, 而在网上没有找到满意的控件。经过研究摸索, 借助于一些成熟的图片类, 终于用 VB 实现了按钮控件。它不仅能够载入常见的格式图片, 如 bmp、gif、jpg、ico、wmf、emf、cur 等, 还能支持 png 格式。主要功能有:

- (1) 支持最多 4 种状态图片 (鼠标在上、按下、移出、按钮无效, 图片可以是横向或纵向) 和透明度; 支持图片载入保存。
  - (2) 支持常用鼠标事件, 还有鼠标移出。
  - (3) 支持 4 种按钮文字颜色, 没有图片时实现文字按钮功能。
- 部分图片格式测试效果如图 1 所示。



图 1 部分图片格式测试效果

## 2 技术难点

(1) 图片透明。绘制 png 图片按钮时有灰色背景, 有人用 maskcolor 会出现白色边缘不清晰。要设置 BackStyle = 0, ClipBehavior=0, Windowless = True。

(2) 4 种状态。一个图片 (横向的或纵向) 有 4 块, 首先要判断按钮处于哪种状态, 再根据状态绘制相应块的图片。如图 2 所示。

(3) 鼠标移出。用定时器不断矩形判断鼠标是否移出, 涉及到的难点就是坐标转换。屏幕坐标用点来表示, ScaleMode 为 vbPixels 是个不错的选择。设计时需要设置窗体、父控件的 ScaleMode, 还有控件为 frame 只能是 twip。另外 Windowless

为 True 时不能获得按钮左上角坐标, 只能获得父控件左上角坐标。本文设置控件 ScaleMode 和父控件 ScaleMode 一样, 利用 GetCursorPos 得到鼠标位置, WindowFromPoint 获得父控件位置, 用 ClientToScreen 转换成屏幕坐标, 配合 hittest 中的 x,y 就可以判断了。示意图如图 3 所示。



图 2 横向和纵向

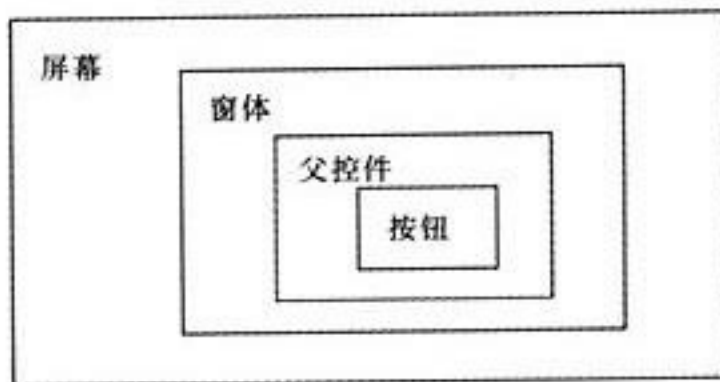


图 3 鼠标坐标获取示意

## 3 关键代码

打开 VB, 点击菜单“文件”→“新建工程”→“Activex 控件”。设置 Windowless = True, BackStyle = 0, ClipBehavior = 0。加一定时器, interval=3。关键代码如下:

```
Public Sub OutTime() ' 此函数在定时器中调用, 用于判断鼠标是否移出
    Dim R As RECT
    Dim pt As POINTAPI, wt As POINTAPI ' 鼠标位置
    GetCursorPos pt
    If WindowFromPoint (pt.X, pt.Y) = UserControl.ContainerHwnd Then ' 找到父控件
        ClientToScreen UserControl.ContainerHwnd, wt ' 容器左上角
        pt.X = pt.X - wt.X - cTopLeftPos.X ' cTopLeftPos 在 'hittest 中计算
```



```

pt.Y = pt.Y - wt.Y - cTopLeftPos.Y
SetRect R, 0&, 0&, UserControl.ScaleWidth, UserControl.
ScaleHeight
If PtInRect(R, pt.X, pt.Y) = 0 Then ' 鼠标移出
    IsOver = False
    IsMouseDown = False
    RaiseEvent MouseOut
    DrawButton 0 ' 绘制正常图片
    Timer1.Enabled = False
    Refresh
    DoEvents
End If
End If
End Sub
Private Sub UserControl_HitTest (X As Single, Y As Single,
HitResult As Integer)
Static hCursor As Long
Dim mPoint As POINTAPI
If Not Ambient.UserMode Then ' 处理设计时点选拖动问题
    HitResult = vbHitResultHit
Else ' 处理运行时鼠标进入事件
    If HitResult = vbHitResultOutside Then
        HitResult = vbHitResultHit
        If UserControl.Enabled Then
            GetCursorPos cTopLeftPos
            ClientToScreen UserControl.ContainerHwnd, mPoint
            cTopLeftPos.X = cTopLeftPos.X - mPoint.X - X
            cTopLeftPos.Y = cTopLeftPos.Y - mPoint.Y - Y
        End If
    End If
End If
End Sub
Public Sub DrawButton (Optional ByVal mValue As Integer =
-1) ' 状态改变时绘制按钮
    If mValue <> DrawCount Then
        If mValue >= 0 Then DrawCount = mValue
        UserControl.Cls
        ButtonSkin
    End If
End Sub
Private Sub ButtonSkin() ' 绘制按钮。图片块数 m_EnQuYu
'Count 可以为 1、2、3、4。
On Error Resume Next
If picPng Is Nothing Then Exit Sub
With UserControl
    RaiseEvent PaintBei ' 自定义事件
    m_state = DrawCount
    If DrawCount < 0 Then
        DrawCount = 0
    ElseIf DrawCount > m_EnQuYuCount - 1 Then
        DrawCount = m_EnQuYuCount - 1
    End If

```

```

    If .Enabled = False Then DrawCount = 3 ' 按钮无效时绘制
    DrawCount = DrawCount Mod m_EnQuYuCount ' 图片不
    ' 足四块时
    If Not (picPng Is Nothing) Then
        If m_EnErectPaint = False Then ' 图片横向
            If m_EnAutoSize Then ' 允许缩放
                picPng.Render .hDC, mRe.Left, mRe.Top, mRe.
                Right - mRe.Left, mRe.Bottom - mRe.Top, picPng.Width *
                DrawCount / m_EnQuYuCount, 0, picPng.Width /
                m_EnQuYuCount, picPng.Height, cOpacity
            Else
                picPng.Render .hDC, mRe.Left, mRe.Top, picPng.
                Width / m_EnQuYuCount, picPng.Height, picPng.Width *
                DrawCount / m_EnQuYuCount, 0, picPng.Width /
                m_EnQuYuCount, picPng.Height, cOpacity
            End If
        Else ' 图片纵向
            If m_EnAutoSize Then
                picPng.Render .hDC, mRe.Left, mRe.Top, mRe.
                Right - mRe.Left, mRe.Bottom - mRe.Top, 0, picPng.Height
                * DrawCount / m_EnQuYuCount, picPng.Width, picPng.
                Height / m_EnQuYuCount, cOpacity
            Else
                picPng.Render .hDC, mRe.Left, mRe.Top,
                picPng.Width, picPng.Height / m_EnQuYuCount, 0, picPng.
                Height * DrawCount / m_EnQuYuCount, picPng.Width,
                picPng.Height / m_EnQuYuCount, cOpacity
            End If
        End If
        End If
        DrawCaption ' 绘制标题
    End With
End Sub
' 刷新显示的标题字符
Private Sub DrawCaption()
On Error Resume Next
With UserControl
    If .Enabled = False Then m_state = 3
    Select Case m_state ' 四种状态标题颜色
    Case 0: SetTextColor .hDC, m_ForeColor
    Case 1: SetTextColor .hDC, m_foreover
    Case 2: SetTextColor .hDC, m_foredown
    Case 3: SetTextColor .hDC, m_foredisable
    End Select
    Select Case m_align ' 对齐方式, 上中下左右
    Case 0:
        DrawText .hDC, m_EnCaption, LenB (StrConv(m_EnCaption,
        vbFromUnicode)), mRe, 33
    Case 1:
        DrawText .hDC, m_EnCaption, LenB (StrConv(m_EnCaption,
        vbFromUnicode)), mRe, 37
    Case 2:

```



## GRAPHICS AND IMAGE PROCESSING

```
DrawText .hDC, m_EnCaption, LenB (StrConv(m_EnCaption,
vbFromUnicode)), mRe, 41
```

```
Case 3:
```

```
DrawText .hDC, m_EnCaption, LenB (StrConv(m_EnCaption,
vbFromUnicode)), mRe, 36
```

```
Case 4:
```

```
DrawText .hDC, m_EnCaption, LenB (StrConv(m_EnCaption,
vbFromUnicode)), mRe, 38
```

```
End Select
```

```
End With
```

```
End Sub
```

```
Public Property Get Imagen() As Byte() ' 图片流的获取
```

```
Dim Test As Boolean
```

```
Dim mBytes() As Byte
```

```
Test = picPng.SaveToStream(mBytes())
```

```
Imagen = mBytes()
```

```
End Property
```

```
Public Property Let Imagen(mB() As Byte)
```

```
Call picPng.LoadPicture_Stream(mB())
```

```
PropertyChanged "PICPNG"
```

```
UserControl.Refresh
```

```
End Property
```

```
' 为用户控件初始化属性
```

```
Private Sub UserControl_InitProperties()
```

```
' Windowless = True ' 设计时设置该属性
```

```
BackStyle = 0 ' 设置控件背景透明
```

```
' ClipBehavior = 0 ' 设置控件剪切方式为无(即全部)
```

```
m_EnAutoSize = False ' 不自动缩放
```

```
m_EnErectPaint = False ' 图片不是纵向
```

```
m_EnQuYuCount = 4 ' 默认是四块
```

```
m_align = 0 ' 对齐方式
```

```
m_EnCaption = "png 按钮" ' 标题
```

```
cOpacity = 100& ' 透明度
```

```
On Error Resume Next ' 和父控件 ScaleMode 一样
```

```
cScaleMode = UserControl.Parent.ScaleMode
```

```
If Err Then
```

```
Err.Clear
```

```
cScaleMode = vbContainerPosition
```

```
End If
```

```
End Sub
```

点击菜单“工程”→“添加属性”。用来载入图片。界面设计如图4所示。

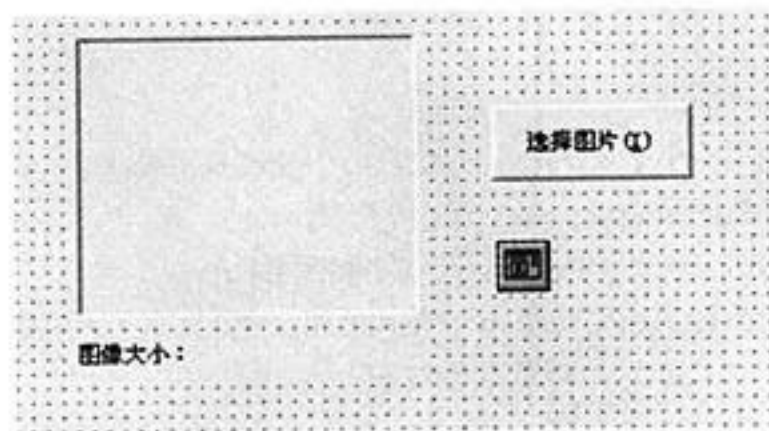


图4 属性页

代码如下：

```
Dim mImagen As New c32bppDIB
```

```
Private Sub Importar()
```

```
Dim mB() As Byte
```

```
With SelectedControls(0)
```

```
Test = mImagen.SaveToStream(mB())
```

```
.Imagen = mB()
```

```
End With
```

```
End Sub
```

```
Private Function GetImageFromBrowser() As String ' 选择文  
' 件并导入
```

```
Dim mB() As Byte
```

```
With CommonDialog1
```

```
.Flags = cdlOFNExplorer Or cdlOFNFileMustExist
```

```
.Filter = "Images|*.bmp;*.gif;*.jpg;*.ico;*.png;*.wmf;*.  
emf;*.cur;*.jpeg" & _
```

```
"Bitmaps|*.bmp|GIF|*.gif|Icon|*.ico;*.cur|JPG|*.jpg;*.jpeg" & _  
"Metafile|*.wmf;*.emf|PNG|*.png"
```

```
.CancelError = True
```

```
End With
```

```
On Error GoTo ExitRoutine
```

```
CommonDialog1.ShowOpen
```

```
sFile = CommonDialog1.FileName
```

```
If sFile <> "" Then
```

```
mImagen.LoadPicture_File (sFile)
```

```
mImagen.Render Picture1.hDC
```

```
Label1.Caption = "图像大小:" & Str(mImagen.Width) & ""  
& Str(mImagen.Height)
```

```
With SelectedControls(0)
```

```
Test = mImagen.SaveToStream(mB())
```

```
.Imagen = mB()
```

```
End With
```

```
GetImageFromBrowser = sFile
```

```
Else
```

```
GetImageFromBrowser = ""
```

```
End If
```

```
mImagen = Nothing
```

```
ExitRoutine:
```

```
End Function
```

```
Private Sub cmdlImportar_Click() ' 选择图片按钮
```

```
Call GetImageFromBrowser
```

```
End Sub
```

```
Private Sub PropertyPage_Paint()
```

```
mImagen.Render Picture1.hDC
```

```
End Sub
```

其他导入模块和类模块不再赘述，具体请看源代码，下载地址为 <http://caoweizhong.ys168.com>。运行环境为 Windows XP+VB6。

(收稿日期：2012-12-13)



# C# 语言开发贪吃蛇游戏

王文举

**摘 要:** 介绍利用 C# 语言开发贪吃蛇游戏的设计方法和代码实现, 对开发类似的 GDI+ 游戏有很好的借鉴作用。

**关键词:** C# 语言; 贪吃蛇游戏; GDI+ 游戏

## 1 引言

贪吃蛇游戏是一种大家较为熟悉的游戏, 通过控制贪吃蛇的移动方向吃掉食物, 而避免蛇头碰到蛇身和背景边界, 如图 1 所示。

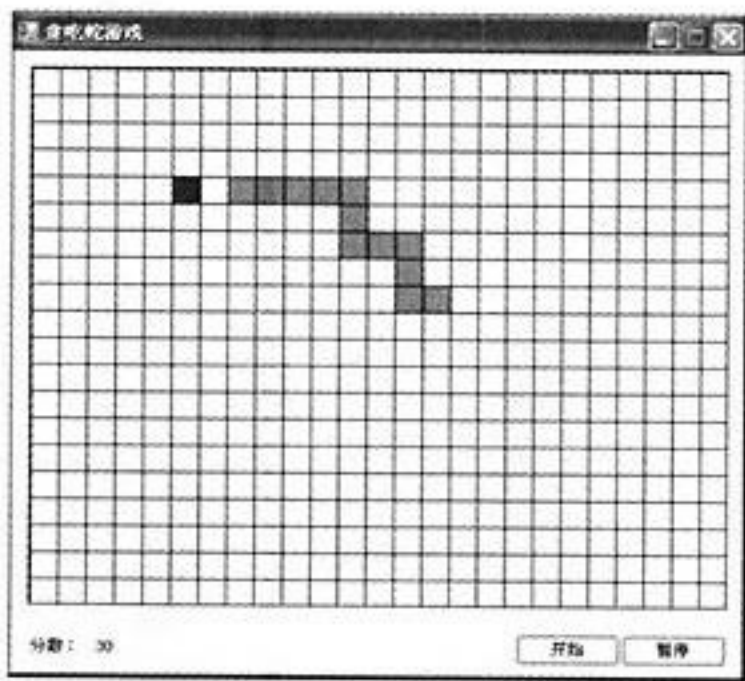


图 1 贪吃蛇游戏界面示例

## 2 设计思路

利用 C# 语言开发了贪吃蛇游戏, 如图 1 所示, 设计中借鉴了 MVC 的设计思路, 除了窗口界面类外设计了 3 个类。

一是模型类 (SnakeMode), 功能为实现模型的构造 (SnakeMode()), 获取贪吃蛇移动后的蛇头位置 (Point SnakeMove (int n)), 生成食物 (ProductFood()), 吃掉食物 (EatFood())。

二是游戏规则类 (Rule), 功能为判断蛇头有无碰到蛇身和背景边界 (bool MoveRule (Point GameP))。

三是画图类 (Draw), 功能为绘制贪吃蛇 (DrawSnake()), 绘制食物 (DrawFood()), 重绘贪吃蛇 (DrawSnakeMove (Point mp)), 绘制游戏场景 (DrawBackground()), 清空游戏背景 (BackgroundClear())。

该方法通过模型、控制、视图相分离, 使程序流程更加清

晰, 并且具有强内聚、松耦合特性, 使功能扩展更加方便。该方法对于开发 GDI+ 游戏有很好的借鉴作用。

## 3 实现代码

### 3.1 窗口界面类

```
public partial class Form1 : Form
{
    SnakeMode mySnake; Draw myDraw;
    Rule myRule; Point nextPoint;
    public static bool GameStart = false; //判断是否开始
    public static bool GameOver = false; //游戏是否结束
    public Form1()
    {
        InitializeComponent();
        timer1.Enabled = false; //停止计时
    }
    //“开始”按键的响应事件
    private void button1_Click(object sender, EventArgs e)
    {
        myRule = new Rule(pictureBox1);
        mySnake = new SnakeMode();
        myDraw = new Draw(pictureBox1);
        myDraw.BackgroundClear(); //清空游戏场地
        myDraw.DrawBackground(); //绘制游戏场地
        myDraw.DrawSnake(); //第一次绘制贪吃蛇
        myDraw.DrawFood(); //绘制食物
        SnakeMode.label = label2;
        GameStart = true; //开始游戏
        GameOver = false;
        textBox1.Focus(); //获取焦点, 绑定键盘事件
        timer1.Start(); //启动计时器
        label2.Text = "0"; //显示当前分数
    }
    //键盘响应事件
    private void Form1_KeyDown(object sender, KeyEventArgs e)
    {
        if (GameStart == true) //开始游戏
        {
            int tem_n = -1; //记录移动键值
            if (e.KeyCode == Keys.Right) //如果按→键
                tem_n = 0; //向右移
            if (e.KeyCode == Keys.Left) //如果按←键
                tem_n = 1; //向左移
```





## GAME PROGRAM

```

        if (e.KeyCode == Keys.Up)//如果按↑键
            tem_n = 2;//向上移
        if (e.KeyCode == Keys.Down)//如果按↓键
            tem_n = 3;//向下移
        if (tem_n != -1 && tem_n != SnakeMode.
direction)//如果移动的方向不是相同方向
        { if (GameOver == false)
            { //如果移动的方向不是相反的方向
                if (!((tem_n == 0 && SnakeMode.direction
== 1 || tem_n == 1 && SnakeMode.direction == 0) || (tem_n
== 2 && SnakeMode.direction == 3 ||
tem_n == 3 && SnakeMode.direction == 2)))
                { SnakeMode.direction = tem_n;//记录移动的方向
                    nextPoint = mySnake.SnakeMove
(tem_n);//获取贪吃蛇移动的下个点
                    SnakeMove(nextPoint);//贪吃蛇运动的结果
                } } }
    }

//计时器事件
private void timer1_Tick(object sender, EventArgs e)
{ nextPoint = mySnake.SnakeMove (SnakeMode.
direction);//获取贪吃蛇移动的下个点
    SnakeMove(nextPoint);//贪吃蛇运动的结果
}
//贪吃蛇运动的结果
private void SnakeMove(Point nextPoint)
{ if (! myRule.MoveRule(nextPoint))//如果移动有效
    { mySnake.EatFood();//遇到食物就吃
        myDraw.DrawFood();//绘制食物
        myDraw.DrawSnakeMove(nextPoint);//绘制贪吃
//蛇下个状态
    }
    else
    { GameOver = true;//游戏结束
        timer1.Stop();//停止记时器
        myDraw.DrawStop();//绘制游戏结束
    }
}
//“暂停”按键的响应事件
private void button2_Click(object sender, EventArgs e)
{ if (button2.Text == "暂停")
    { timer1.Stop();//暂停
        GameStart = false;//开始游戏
        button2.Text = "继续";
        textBox1.Focus();//获取焦点
    }
    else
    { timer1.Start();//继续
        GameStart = true;
        button2.Text = "暂停";
        textBox1.Focus();//获取焦点
    }
}

```

```

    }
}
}

3.2 SnakeMode 类
class SnakeMode
{ public static int Node = 20;//设置每个骨节的大小
    public static int direction = 0;//设置方向
    public static Point[] Place = new Point[6];//设置每个骨
//节的位置
    public static ArrayList List = new ArrayList();//实例化
//ArrayList 数组
    public static Point Food = new Point(-1, -1);//设置食物
//的所在点
    public static bool haveFood = false;//是否有食物
    public static Label label;//记录 Label 控件
    public SnakeMode()
    { for (int i = 0; i < Place.Length; i++)//绘制贪吃蛇
        { Place[i].X = (Place.Length - i - 1) * Node;//设置
//骨节的横坐标位置
            Place[i].Y = (Rule.Field_Height / 2) - Node;//设置
//骨节的纵坐标位置
        }
        List = new ArrayList(Place);//记录每个骨节的位置
        direction = 0;
        haveFood = false;
        ProductFood();//生成食物
    }
//获取贪吃蛇运动的下个点
    public Point SnakeMove(int n)
    { Point tem_Point = new Point(-1, -1);//定义坐标结构
        switch (n)
        { case 0://右移
            { tem_Point.X = ((Point)List[0]).X + Node;//蛇头向右移
                tem_Point.Y = ((Point)List[0]).Y; break; }
            case 1://左移
            { tem_Point.X = ((Point)List[0]).X - Node;//蛇头向左移
                tem_Point.Y = ((Point)List[0]).Y; break; }
            case 2://上移
            { tem_Point.Y = ((Point)List[0]).Y - Node;//蛇头向上移
                tem_Point.X = ((Point)List[0]).X; break; }
            case 3://下移
            { tem_Point.Y = ((Point)List[0]).Y + Node;//蛇头向下移
                tem_Point.X = ((Point)List[0]).X; break; }
        }
        return tem_Point;
    }
//生成食物
    public void ProductFood()
    { if (haveFood == false)//如果没有食物
        { Point tem_p = new Point(-1, -1);//定义坐标结构
            bool tem_bool = true;//是否计算出食物的合位置

```



```

bool tem_b = false;//判断食物是否和蛇身重叠
while (tem_bool)//计算食物的显示位置
{
    tem_b = false;
    tem_p = FoodPlace();//随机生成食物的位置
    for (int i = 0; i < List.Count; i++)//遍历整个蛇
//身的位置
//蛇身重叠
    {
        if (((Point)List[i]) == tem_p)//如果食物是否和
        {
            tem_b = true;//记录重叠
            break;
        }
        if (tem_b == false)//如果没有重叠
            tem_bool = false;//退出循环
    }
    Food = tem_p;//记录食物的显示位置
}
haveFood = true;//有食物
}
//随机生成食物的节点
public Point FoodPlace()
{
    int tem_W = Rule.Field_width / 20;//获取场地的行数
    int tem_H = Rule.Field_Height / 20;//获取场地的列数
    tem_W = RandW.Next(0, tem_W - 1);//生成食物的
//横向坐标
    Random RandH = new Random();//实例化 Random 类
    tem_H = RandH.Next(0, tem_H - 1);//生成食物的纵
//向坐标
    Point tme_P = new Point (tem_W * Node, tem_H *
Node);//生成食物的显示位置
    return tme_P;
}
//蛇头碰到食物,吃掉食物,并加分
public void EatFood()
{
    if (((Point)List[0]) == Food)//如果蛇头吃到了食物
{
    List.Add(List[List.Count - 1]);//在蛇的尾部添加蛇身
    haveFood = false;//没有食物
    ProductFood();//生成食物
    label.Text = Convert.ToString (Convert.ToInt32
(label.Text) + 5);//显示当前分数
}
}
}
}

```

## 3.3 Rule 类

```

class Rule
{
    public static int Field_width = 0;//场地的宽度
    public static int Field_Height = 0;//场地的高度
    Control Mycontrol;
    public Rule(Control _mycontrol)
    {
        Mycontrol = _mycontrol;
        Field_width = Mycontrol.Width;//获取场地的宽度
        Field_Height = Mycontrol.Height;//获取场地的高度
    }
}

```

```

}
//运动是否有效,蛇头有无碰到蛇身,有无出界
public bool MoveRule(Point GameP)
{
    bool gameOver = false;//游戏是否结束
    bool tem_body = false;//记录蛇身是否重叠
    for (int i = 1; i < SnakeMode.List.Count; i++)//遍历
//所有骨节
    {
        if (((Point)SnakeMode.List [0]) == ((Point)
SnakeMode.List[i]))//如果蛇头碰到蛇身
            tem_body = true;//游戏失败
    }
    //判断蛇头是否超出游戏场地
    if (GameP.X <= -20 || GameP.X >= Field_width - 1
|| GameP.Y <= -20 ||
        GameP.Y >= Field_Height - 1 || tem_body)
    {
        gameOver = true;//游戏失败
    }
    return gameOver;
}
}

```

## 3.4 Draw 类

```

class Draw
{
    Control Mycontrol; Graphics g;
    public static SolidBrush SolidB = new SolidBrush(Color.
Red);//设置贪吃蛇身体的颜色
    public static SolidBrush SolidF = new SolidBrush(Color.
Blue);//设置食物的颜色
    public static SolidBrush SolidD;//设置背景颜色
    public Draw(Control control)
    {
        Mycontrol = control;
        g = control.CreateGraphics();//创建背景控件的 Graphics 类
        SolidD = new SolidBrush(Mycontrol.BackColor);
    }
    //第一次绘制贪吃蛇
    public void DrawSnake()
    {
        for (int i = 0; i < SnakeMode.List.Count; i++)
        {
            g.FillRectangle(SolidB, SnakeMode.Place[i].X + 1,
SnakeMode.Place [i].Y + 1, SnakeMode.Node - 1,
SnakeMode.Node - 1);//绘制骨节
        }
    }
    //绘制食物
    public void DrawFood()
    {
        g.FillRectangle (SolidF, SnakeMode.Food.X + 1,
SnakeMode.Food.Y + 1,
        SnakeMode.Node - 1, SnakeMode.Node - 1);
    }
    //绘制游戏结束
    public void DrawStop()
    {
        g.DrawString("游戏结束!", new Font("宋体", 50,
FontStyle.Bold),
(下转第 81 页)
    }
}

```





# 手机基站定位的实现与应用

汪永松

**摘要:** 从开发者的角度介绍了在 Android 平台基于手机基站实现低成本的定位服务, 并通过开发实例, 对此定位的应用进行了详细说明。

**关键词:** Android 系统; 手机基站定位; J2EE 平台; Web 地图 API

## 1 概述

### 1.1 基于手机基站定位应用实例

从事过导航应用的读者应该比较了解, 定位应用的模式可分为 3 个基本部分:

(1) 获取移动对象的位置信息 (通常用经纬度坐标表示), 读者可以理解为绝对位置 (全球定位信息);

(2) 与地图 (通常包含: 背景、道路、叠加文字和兴趣点等图层) 进行匹配并定位, 读者可以理解为相对位置 (在某栋大楼里、在某条街道上);

(3) 基于地图中移动对象位置点的关联性分析, 例如: 周边存在哪些服务设施、相同位置附件还存在哪些移动对象群体等。

图 1 所示的是在网页地图中以叠加图的形式展示某手机设备的全天位置轨迹, 当光标移动在某叠加图上方时, 会显示该位置记录时点的提示文本。

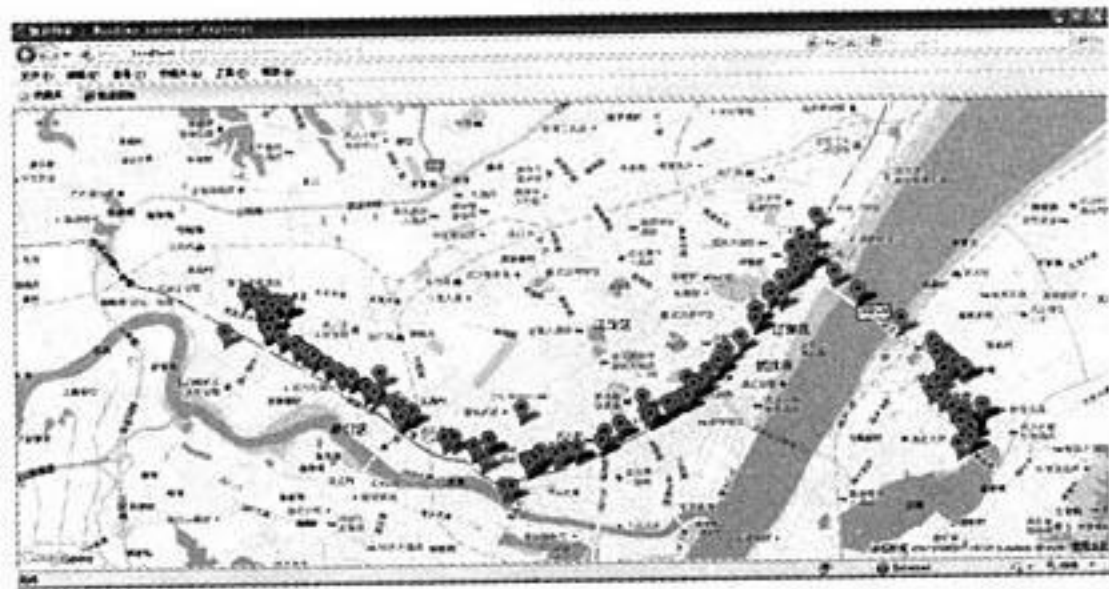


图 1 手机定位轨迹回放

结合前文对定位应用模式的分析, 读者可以看出, 图 1 中包含了地图、手机设备的位置信息和记录时点信息。通过该平台, 用户可以知道指定手机在某个时间点在哪里, 或在某个位置的时间点。将这些关键的基本信息提取到业务系统中即可实现定位服务与现有业务的集成, 从而会使业务平台中数据的客观性得到较大程度的加强。

### 1.2 关键技术

从前文分析可知, 手机定位应用需要获取的基本要素包括: 手机设备位置信息和背景地图。在手机平台, 基于手机基站的位置信息需要通过手机的电话系统来获取; 而该位置信息是手机基站的蜂窝定位信息, 而不是直接被用户所识别的经纬度信息, 其还需要通过位置服务接口进行一次转换; 而背景地图一般通过网页地图 API 进行调用。所以, 基于手机基站定位的关键技术包括: 手机平台的电话系统、位置服务接口通信和网页地图 API。

#### 1.2.1 手机平台电话系统

电话系统是手机平台最为核心的部分, 在 Android 平台, 电话系统不仅提供了拨打/接听呼叫的功能, 而且还提供了丰富的用于获取手机网络以及电话状态的 API。通过电话管理器接口 (TelephonyManager) 不仅可以获取手机服务网络、SIM 卡状态、电话类型、数据状态和呼叫状态等, 而且还可以获取手机的蜂窝位置, 如表 1 所示。

表 1 Android 平台电话系统信息类型

方法	说明
getCallState ()	获取呼叫状态
getCellLocation ()	获取蜂窝位置
getDeviceId ()	获取设备标识
getLine1Number ()	获取线路 1 号码
getNetworkCountryIso ()	获取网络国家代码
getNetworkOperator ()	获取网络运营商代码
getNetworkType ()	获取网络类型
getPhoneType ()	获取电话类型
getSimCountryIso ()	获取 SIM 卡国家代码
getSimOperator ()	获取 SIM 运营商代码
getSimSerialNumber ()	获取 SIM 序列号
getSimState ()	获取 SIM 状态





除此之外，Android 平台的电话系统还提供了侦听电话状态的侦听器，通过侦听器可以捕捉电话状态的变化，例如：信号强弱、呼叫状态、蜂窝位置、数据状态、服务状态等，如表 2 所示。

表 2 Android 平台电话状态侦听类型

标识	说明
LISTEN_CALL_STATE	侦听设备呼叫状态的变化
LISTEN_CELL_LOCATION	侦听设备蜂窝位置的变化
LISTEN_DATA_ACTIVITY	侦听数据连接中数据交通方向的变化
LISTEN_DATA_CONNECTION_STATE	侦听数据连接状态的变化
LISTEN_NONE	停止侦听
LISTEN_SERVICE_STATE	侦听网络服务状态的变化
LISTEN_SIGNAL_STRENGTHS	侦听网络信号强度的变化

由此可知，通过手机平台的电话系统，开发者可以获取设备当前的蜂窝位置并侦听其位置的变化。但是，该蜂窝位置是由手机服务网络定义的区域代码，并不是地理经纬度坐标，如图 2 所示的是手机基站定位信息的结构。

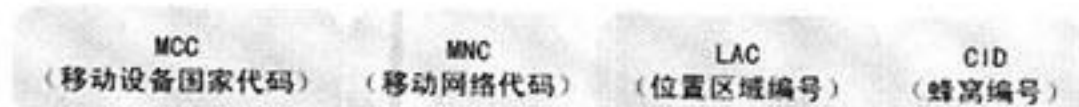


图 2 手机基站定位信息结构

按照国际标准，中国的 MCC 为 460；中国移动的网络代码为：00、02 和 07。位置区域编号和蜂窝编号是网络运营商按照基站序号进行的编码。因为是编码，所以无法通过基站位置信息直接计算出其对应的经纬度坐标，而是需要向位置服务接口请求转换坐标。

## 1.2.2 位置服务接口通信

此处所用到的位置服务接口是一个 Web 服务，其使用 JSON 作为数据交换格式。该服务接收手机网络的蜂窝位置，返回其所对应的经纬度坐标。其通信采用 HTTP 方式，连接成功之后通过连接的写/读接口来写入请求或读取回复。

## 1.2.3 地图 API

对于绝大多数应用而言，光有一大串的经纬度坐标点并没有什么用途，坐标点往往需要与背景地图进行匹配，甚至与该地图中该坐标附近的 POI 产生关联，例如该坐标点附近的有关设施等。

在越来越多的应用中，使用背景地图的最佳方式是使用网页地图，而使用网页地图实际上就是使用该品牌网页地图的 API。通过地图 API，开发者可以将地图显示在指定的网页组件中，可以添加地图叠加图、添加路径等。

如此一来，开发者可以借助网页地图将手机设备的一大串经纬度坐标在背景地图上显示，通过地图匹配，可以获得

某时间点手机设备的位置，而通过时间戳以及位置信息即可与业务应用系统进行关联，从而实现手机定位技术与应用系统的融合。

本文中使用的是谷歌地图 JavaScript API 第三版，详细参考可以访问谷歌开发者网站有关地图的页面：<https://developers.google.com/maps/>。

## 2 设计过程

### 2.1 开发思路

通过以上的介绍，读者应该初步了解基于手机基站定位应用的大致思路如下：

- (1) 使用手机平台的电话系统获取手机的基站定位信息。
- (2) 通过位置服务接口使用手机基站定位信息获取对应的经纬度信息。
- (3) 使用地图 API 将手机经纬度位置点在 Web 地图上进行展示。
- (4) 使用手机的经纬度位置与参考位置点进行关联性分析。

### 2.2 业务流程

依据低成本的原则，暂不考虑手机能够长时与 Wi-Fi 或 3G 网络保持连接，而是间断性连接。首先，手机的定位信息将以数据文件的形式先保存到 SD 卡中；当手机接入 Wi-Fi 网络后，将数据文件上传到服务器；位置查询模块将从数据文件中抽取手机基站定位信息，并向远程定位服务器查询对应的经纬度信息；最后所有的定位信息都将记录到数据中心；然后基于数据中心通过 Web 服务的方式向桌面系统或移动端提供定位轨迹，如图 3 所示。

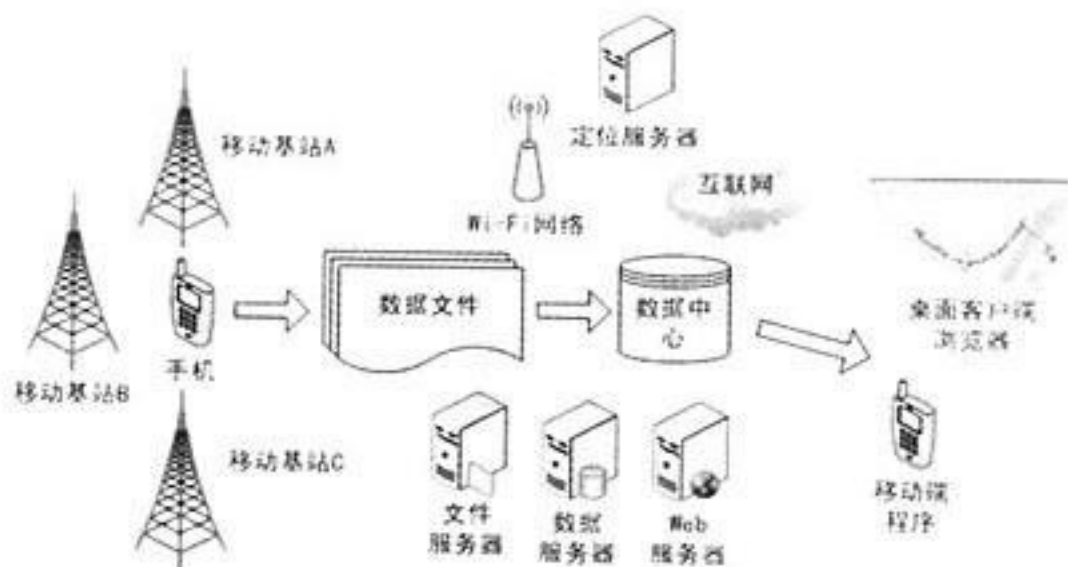


图 3 手机基站定位业务流程

在示例案例中，分为 4 个部分来实现整个定位应用：第一部分在手机端，用户侦听并记录手机设备基于基站的位置信息，保存到 SD 卡中；第二部分别部署在手机端和文件服务器端，用于将手机 SD 卡中的数据文件上传到指定文件服务器中；第三部分部署在应用服务器端，用于抽取手机设备的基站定位记录，再向远程定位服务器进行查询，最后将所有的定位信息都存入到数据库中；第四部分部署在 Web 服务器，以 Web 服



## COMPUTER SECURITY AND MAINTENANCE

服务器的形式向桌面系统或移动终端提供定位信息服务。

## 2.3 主要类设计

## 2.3.1 位置跟踪服务—TrackService

该类被设计为一个服务 (Service)，用于在后台持续地侦听手机设备的基站定位信息，即：移动设备国家代码、移动网络代码、位置区域编号和蜂窝单元编号，并将信息写入到 SD 卡上的数据文件中。

为了区分不同的手机设备和日期，数据文件名的格式定为：“<用户标识>\_<日期>.txt”，其中用户标识可以是手机号码或 IMEI；日期的格式为：“yyyymmdd”，如图 4 所示。



图 4 基站定位数据文件存放形式

记录格式定为以：“<时间戳>;MCC;MNC;LAC;CID”，其中时间戳的格式为：“hhmmss”，如图 5 所示。



图 5 基站定位数据文件内容

此外，为了自动进行侦听，还使用广播接收器 (Broadcast Receiver) 来接收设备启动完毕的广播，当手机启动完毕后将自动启动服务。

## 2.3.2 定位查询桩—LocationQueryStub

定位查询桩类完成 3 个主要任务：

- (1) 提取基站定位信息记录。即从数据文件中逐条读取基站定位记录，并存入到内存记录数组中（及下文中的位置信息）。
- (2) 调用位置查询线程查询基站定位信息对应的经纬度信息。即将位置信息中的基站定位信息通过 HTTP 的方式向定位服务进行查询，并将查询得到的经纬度信息回填到该位置信息记录。

(3) 将查询结果存入到数据库。即将填充完整的位置信息记录存入到数据库。为了考虑将定位查询功能移植到 Android 平台，存储数据库采用 SQLite 数据库，该数据库能够方便地从桌面平台转移到 Android 平台中。

## 2.3.3 位置信息—Location

位置信息包含了手机设备完整的位置信息，其内容如表 3 所示。

表 3 位置信息构成内容

内容	说明
日期	格式为“yyyymmdd”（非空）
时间	格式为“hhmmss”（非空）
电话设备标识	电话设备标识（电话号码或 IMEI 号码）（非空）
位置区域信息 (LAI)	MCC, MNC, LAC, CID（非空）
纬度	乘以 105 的整数值（非空）
经度	乘以 105 的整数值（非空）
街道	街道信息（可为空）
区域	区域信息（可为空）
城市	城市信息（可为空）
国家	国家信息（可为空）

通过表 3 的各项，读者可以获知指定手机设备在某一时间点的地理位置坐标信息。

## 2.3.4 位置查询线程—LocQueryThread2

位置查询线程调用位置查询服务，以位置区域信息 (LAI) 作为输入参数，获取服务返回的经纬度信息。实例中的位置查询服务都是由谷歌公司提供，笔者在项目中尝试过两个服务接口。第一个服务的 URL 为“http://www.google.com/loc/json”，其请求及返回以 JSON 格式进行编码；第二个服务的 URL 为“http://www.google.com/glm/mmap”，其请求及返回以 HTTP 字节流的方式进行组织。显而易见，第一个服务接口的调用要比第二个容易得多，因为 JSON 格式不仅易于理解，而且其内容组织和解析都有标准的工具库可用；而字节流的组织和解析必须按照数据项的顺序和大小进行底层的字节控制，其复杂度要比 JSON 格式的方式要高得多。但是经过测试，第二个服务接口的查询效果要比第一个稳定得多，不仅查询效率较高，而且大部分查询都可以获取有效的返回值；相比之下，采用 JSON 方式的服务接口的查询效率要比使用字节流的方式要低一些，不仅如此，其查询得到有效返回值的次数要比第二个服务接口少得多（对于相同的记录）。所以，笔者推荐采用第二个服务接口。

因为考虑到网络访问瓶颈的因素，为了不造成主线程的阻塞，所以位置查询模块采用线程的方式。其大致流程为：首先通过 URL 建立 HTTP 连接，然后通过往连接的输出流写入参数来实现发送请求，之后通过读取连接的输入流来获取返回结果。

为了优化查询效率，在网络查询之前先在数据库中进行检索，如果该位置区域信息有对应的查询结果（经纬度）则直接使用其结果，无需再次进行网络查询。





## 2.3.5 查询结果以叠加图显示—query\_markers.jsp

查询结果的是以网页的形式 (JSP 页面) 进行展示。该 JSP 从数据中心获取指定条件 (按照设备和日期) 的记录集 (经纬度坐标和时间戳), 然后按照网页地图 API 的调用方式 (案例中采用的是 JavaScript 方式) 将网页地图 API 和位置记录集组织成页面内容。

案例中采用的是谷歌地图, 地图 API 是谷歌地图 JavaScript API 第 3 版。其中基础的功能要求是: (1) 显示背景地图; (2) 在各位置点上添加叠加图标。

## 2.4 数据库设计

案例中使用的是 SQLite 数据库, 其中两个主要数据表为轨迹列表和位置表。

### 2.4.1 轨迹列表 (tab\_track\_list)

轨迹列表用于存储用户 (手机设备) 的轨迹历史 (按天), 如表 4 所示。

表 4 轨迹列表数据定义

字段名	类型	约束	默认值	备注
_id	INTEGER	PK		自增
user_id	TEXT			用户 ID (电话号码或 IMEI)
_date	TEXT			日期

### 2.4.2 位置表 (tab\_sdl\_locations)

位置表用于存储构成轨迹的每一个点的位置信息, 如表 5 所示。

表 5 位置表数据定义

字段名	类型	约束	默认值	备注
_id	INTEGER	PK		自增
track_id	INTEGER	FK		参考轨迹列表中的主键 ID
_time	TEXT			时间点
mcc	INTEGER			移动设备国家代码
mnc	INTEGER			移动网络代码
lac	INTEGER			位置区域编号
cid	INTEGER			蜂窝单元编号
lat	INTEGER			纬度 (乘以 105 的整数值)
lon	INTEGER			经度 (乘以 105 的整数值)
street	TEXT			街道
region	TEXT			区域
city	TEXT			城市
country	TEXT			国家

根据以上两个数据表的定义, 可以获取指定用户 (手机设备) 在某一天的位置轨迹 (按照时间点排序)。

## 3 开发过程

### 3.1 获取手机设备基站定位信息

#### 3.1.1 初始化电话系统管理器

为了获取手机基站定位信息, 必须获取电话系统服务管理器实例, 并设置状态侦听器, 如代码 1 所示:

```
//代码 1 初始化电话系统
//文件名: TrackService.java
//获取电话信息服务管理器
final String sname = Context.TELEPHONY_SERVICE;
mTelMgr = (TelephonyManager)this.getSystemService(sname);
//初始化侦听器 (该类的定义参见代码 2)
mListener = new FooPhoneStateListener();
//设置位置侦听
if(mTelMgr != null) {
    mTelMgr.listen (mListener, PhoneStateListener.LISTEN_NONE);
    mTelMgr.listen (mListener, PhoneStateListener.LISTEN_CELL_LOCATION);
}
```

代码 1 中的电话状态侦听器是定制类 (FooPhoneStateListener), 该类通过继承电话状态侦听器来实现定制功能。

需要注意的是, 在 Android 平台获取手机基站信息需要在程序清单中声明读取电话状态和存取位置的使用许可, 如下所示:

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

#### 3.1.2 获取基站定位信息

代码 2 中包含了自定义电话状态侦听器的定义以及如何获取基站定位信息:

```
//代码 2 自定义电话状态侦听器
//文件名: TrackService.java
class FooPhoneStateListener extends PhoneStateListener {
    @Override
    public void onCellLocationChanged(CellLocation loc) {
        //判断电话管理器实例是否为空
        if(mTelMgr == null) {
            return;
        }
        updateLocation(mTelMgr, loc);
    }
}
```





## COMPUTER SECURITY AND MAINTENANCE

```

}
//更新位置
public void updateLocation (TelephonyManager mgr,
CellLocation loc) {
    //获取手机当前的位置区域识别信息
    final String operator = mgr.getNetworkOperator();
    final int mcc = Integer.parseInt(operator.substring(0,3));
    final int mnc = Integer.parseInt(operator.substring(3));
    final int lac = ((GsmCellLocation)loc).getLac();
    final int cid = ((GsmCellLocation)loc).getCid();
    //判断定位信息合法性
    if((lac==0)&&(cid==0)) {
        return;
    }
    if( (lac < 0) || (lac >= 0xffff) || (cid < 0) || (cid >= 0xffff) ) {
        return;
    }
    //获取用户标识(电话号码或IMEI)
    String userId = mgr.getLine1Number();
    if(userId.trim().length()<1) {
        userId = mgr.getDeviceId();
    }
    //定位信息记录行
    final String line = ""+mcc+IConfig.SEP+mnc+IConfig.SEP+lac+IConfig.SEP+cid;
    Log.d(IConfig.APP_TAG, "计时器更新:"+line);
    //判断记录是否存在重复
    if( TrackService.this.mLastLine.compareTo(line) != 0) {
        DebugUtil.getInstance().println(userId.trim(),line);
        TrackService.this.mLastLine = line;
        Log.d (IConfig.APP_TAG, " 更新位置:"+
TrackService.this.mLastLine);
    }
};

```

代码2中,自定义电话状态侦听器类通过实现基类的方法(onCellLocationChanged)来获取基站位置的信息。通过电话系统的网络运营商信息可以获得移动设备国家代码(MCC)和移动网络代码(MNC);通过回调方法的参数(CellLocation类)可获取位置区域编码(LAC)和基站单元编码(CID)。

需要注意的是本案例中,获取的是GSM网络的基站定位信息,对于其他网络暂无涉及。

### 3.1.3 存储基站定位信息

代码3是将手机基站定位信息记录存储到手机存储器的主要代码。

```

//代码3 存储基站定位信息记录
//文件名:DebugUtil.java
public void println(String user, String log) {
    //目标文件夹
    final String fileDir = Environment.getExternalStorageDirectory()
+

```

```

File.separator+IConfig.APP_TAG;
File d = new File(fileDir);
if(! d.exists()) {
    d.mkdir();
}
//文件全路径
final String filePath = fileDir+File.separator+user+"_"+
FooSysUtil.getInstance().getDateStr("")+".txt";
try {
    File f = new File(filePath);
    PrintWriter pw = new PrintWriter(new FileWriter(f,
true));
    pw.println(FooSysUtil.getInstance().getTimeStr("")+
IConfig.SEP+log);
    pw.close();
} catch (IOException e) {
    e.printStackTrace();
}
}

```

代码3中,通过数据文件的方式来存储记录。其中首先要确定目标文件夹,通过环境管理器接口可以获取当前外部存储器(SD)的文件夹路径。需要注意的是,在Android平台往SD卡写入数据需要在程序清单中声明写外部存储器的使用许可,如下所示:

```

<uses -permission android:name = "android.permission.
WRITE_EXTERNAL_STORAGE"/>

```

## 3.2 通过位置查询服务接口查询经纬度

### 3.2.1 位置查询桩模块

位置查询桩模块用于驱动对基站定位记录文件的解析、查询基站定位信息对应的经纬度信息以及将查询结果保存到数据中心。代码4是该模块的主要代码。

```

//代码4 位置查询桩模块
//文件名:LocationQueryStub.java
File dir = new File(args[0]);
File[] files = dir.listFiles();
for(int i = 0; i < files.length; ++i) {
    //获取数据文件路径
    final String filePath = files[i].getAbsolutePath();
    //获取日期
    File f = new File(filePath);
    final String fnames[] = f.getName().split("\\.");
    final String infos[] = fnames[0].split("_"); //id_日期
    //判断该文件是否已经上传到数据库
    if(isAlreadyUpload(conn, infos[0], infos[1])==true) {
        DebugUtil.getInstance().println(infos[0]+"的"+infos
[1]+"的记录已经存在!");
        continue;
    }
}
//初始化数据文件解析器

```



```

LogFileParser parser = new LogFileParser(filePath);
//初始化位置查询器
LocationQuery query = new LocationQuery();
for(int j = 0; j < parser.getLocations().size(); ++j) {
    //遍历解析到的基站位置记录
    Location loc = parser.getLocations().get(j);
    //该位置信息已经存在于数据库,则直接设置位置信息
    if(isExists(conn, loc)) {
        setLocation(conn, loc);
    } else { //否则通过网页服务进行查询
        query.query(loc);
    }
}
//将查询后所有记录批量保存到数据库
try {
    conn.setAutoCommit(false);
    saveToDb(conn, parser.getLocations());
    conn.setAutoCommit(true);
} catch (SQLException e) {
    e.printStackTrace();
}
...
}

```

代码 4 中, 桩模块遍历所有基站定位信息记录文件, 并使用解析 (LogFileParser) 器提取文件中的记录行到位置 (Location) 记录数组, 再调用位置查询器 (LocationQuery) 对每条位置记录进行查询, 查询结果将回填到位置信息记录中, 最后将完整的位置记录数组存入到数据库。

其中, 为了提高查询效率, 分别按照文件和记录的级次对查询内容进行过滤。对于已经提取的记录文件将不再进行提取; 对于数据库中已经存在的位置信息, 将不再通过服务查询, 而是直接使用原有查询结果。

### 3.2.2 位置查询线程

代码 4 中, 位置查询所传入的参数是位置记录, 该参数也将用于回填查询结果。为了不影响主线程, 该查询模块采用线程的方式, 代码 5 是该线程的主要代码:

```

//代码 5 位置查询线程
//文件名: LocQueryThread2.java
//查询位置区域识别码所对应的经纬度信息
private void queryLatitudeLongitude (Location loc) throws
IOException {
    //Web 服务地址
    final String urlString = "http://www.google.com/glm/
mmap";
    //HTTP URL 连接接口
    HttpURLConnection httpConn = null;
    try {
        URL url = new URL(urlString);
        URLConnection conn = url.openConnection();

```

```

        httpConn = (HttpURLConnection) conn;
        httpConn.setDoOutput(true);
        httpConn.setDoInput(true);
        httpConn.setDefaultUseCaches(false);
        httpConn.setRequestMethod("POST");
        httpConn.setConnectTimeout(60000);
        httpConn.connect();

        //通过向 HTTP 连接的输出流来发送请求(位置区域
        //信息)
        OutputStream os = httpConn.getOutputStream();
        postData(os, loc.getLai());
        os.close();
        //获取 HTTP 连接的输入流用于读取返回结果
        DataInputStream dis = new DataInputStream
        (httpConn.getInputStream());
        getData(dis, loc);
        // 关闭流
        dis.close();
    } catch (Exception e) {
        System.out.println("解析"+loc.getLai().toString()+"异
        常!");
    } finally {
        // 关闭连接
        if (null != httpConn) {
            httpConn.disconnect();
        }
    }
}

```

通过代码 5 读者不难看出, 查询过程采用 HTTP 连接的方式进行。连接成功后, 通过向连接的输出流写入基站定位信息来发送查询请求; 再通过从连接的输入流读取字节来读取返回结果。

代码 6 是向输出流写入数据的关键代码:

```

//代码 6 发送位置查询请求
//文件名: LocQueryThread2.java
//发送请求
private void postData (OutputStream os, Lai lai) throws
IOException {
    DataOutputStream dos = new DataOutputStream(os);
    dos.writeShort(21);
    dos.writeLong(0);
    dos.writeUTF("en");
    dos.writeUTF("Android");
    dos.writeUTF("1.0");
    dos.writeUTF("Web");
    dos.writeByte(27);
    dos.writeInt(0);
    dos.writeInt(0);
    dos.writeInt(3);
}

```





## COMPUTER SECURITY AND MAINTENANCE

```

dos.writeUTF("");
//写入 CID+LAC+MNC+MCC
dos.writeInt(lai.getCid()); //cid
dos.writeInt(lai.getLac()); //lac
dos.writeInt(lai.getMnc()); //mnc
dos.writeInt(0); //mcc
//dos.writeInt(lai.getMcc()); //mcc
dos.writeInt(0);
dos.writeInt(0);
//提交写入字节
dos.flush();
dos.close();
}

```

代码 6 中, 按照服务接口的要求写入数据项, 其中关键内容包括基站定位信息。

代码 7 是获取位置查询结果的主要代码:

```

//代码 7 获取位置查询请求
//文件名: LocQueryThread2.java
//获取返回结果
private void getData (DataInputStream dis, Location loc)
throws IOException {
    dis.readShort();
    dis.readByte();
    int code = dis.readInt();
    if (code == 0) { //读取并设置纬度和经度
        loc.setLat(dis.readInt());
        loc.setLon(dis.readInt());
        //
        dis.readInt();
        dis.readInt();
        dis.readUTF();
    } else {
        System.out.println("解析"+loc.getLai().toString()+"失败!");
    }
}

```

代码 7 中, 从 HTTP 连接的输入流读取纬度和经度内容并回填到位置记录。至此, 完整的位置记录获取完毕。

### 3.3 手机位置轨迹的展示

#### 3.3.1 轨迹历史列表

为了更好地展示各手机位置轨迹, 笔者将手机历史轨迹按照设备和日期的维度进行归集, 即按照日期对设备的历史轨迹进行浏览。同时, 为了便于浏览, 将对日期列表进行分页, 如代码 8 所示:

```

//代码 8 对指定设备按日期进行分页
//文件名: profile.jsp
<table width="600" cellspacing="0" cellpadding="0" border="1"
bordercolor="#cccccc">
<%
//连接 SQLite 数据库

```

```

final String filePath = baseRealPath + "track_locations.
db";
Connection conn = SQLite3Util.getInstance ().
getConnection(filePath);
if(conn==null) {
    return;
}
//获取设备的日期列表数量并计算分页数
final int rowCount = SQLite3Util.getInstance ().
getResultRows(conn,
    "SELECT count(_id) AS item " +
    "FROM 'tab_track_list' "+
    "WHERE (user_id='"+uld+"');");
final int pageCount = ((rowCount%PAGE_SIZE)! =0)?
    ((rowCount/PAGE_SIZE) +1):
(rowCount/PAGE_SIZE);
//获取当前分页后的日期列表
ArrayList <String> items = SQLite3Util.getInstance ().
getArrayList(conn,
    "SELECT (_date||'|'||_id) AS item " +
    "FROM 'tab_track_list' "+
    "WHERE (user_id='"+uld+"') ORDER BY '_id'
LIMIT "+PAGE_SIZE+
    " OFFSET "+((pageNo-1)*PAGE_SIZE)+";");
if(items==null) {
    return;
}
//关闭数据库连接
try {
    conn.close();
} catch (SQLException e) {
    e.printStackTrace();
}
//输出表格头
out.println ("<tr height='\40\><th width='\100\>序号</
th><th width='\200\>日期</th><th width='\150\>路径模
式</th><th width='\150\>标注模式</th></tr>");
//输出表格行
for(int i = 0; i < items.size(); ++i) {
    //将设备 ID 和日期作为参数通过链接进行传递
    final String item = items.get(i);
    final String[] pair = item.split("\\|");
    out.println("<tr height='\30\><td align='\center\>"+
        (((pageNo-1)*PAGE_SIZE) +i +1) + " </td><td
align='\center\>"+pair[0]+
        "</td><td align='\center\><a href='\query_path.
jsp?track_id='"+pair[1]+
        "\><img src='\images/path.png\> width='\24\>
height='\24\></a></td>
        <td align='\center\><a href='\query_markers.
jsp?track_id='"+
        pair [1]+"'\><img src='\images/marker.png\></

```



```
a></td></tr>");
    }
%>
</table>
```

代码 8 中，首先通过获取用户的日期总数（即某手机存在多少天的记录）确定分页数，再通过分页信息获取当前分页的日期列表，最后以用户标识和日期作为参数添加到表格单元格的链接中，链接模块即可显示指定用户在某日期的所有记录点。图 6 是轨迹历史列表页面实例图。

序号	日期	路径模式	标注模式
1	20120226		
2	20120227		
3	20120228		

图 6 手机轨迹历史列表

### 3.3.2 以叠加图的方式展示轨迹

无论在网页地图中以叠加图还是以路径的方式进行展示，首先必须显示背景地图；而要显示背景地图就必须使用地图 API，如代码 9 所示：

```
//代码 9 地图 API 显示 Web 地图
//文件名:query_markers.jsp
<html DIR="LTR">
<head>
<meta name="viewport" content="width=device-width,
initial-scale=1.0, user-scalable=no" />
<meta http-equiv="content-type" content="text/html;
charset=UTF-8" />
<link href="http://code.google.com/apis/maps/documentation/
javascript/examples/default.css" rel="stylesheet" type="text/
css" />
<title>轨迹回放</title>
<script type="text/javascript" src="http://maps.google.com/
maps/api/js?sensor=false"></script>
<script type="text/javascript">
    function initialize() {
        var myLatLng = new google.maps.LatLng(30.6, 114.26);
        var myOptions = {
            zoom: 13,
            maxZoom: 21,
            minZoom: 1,
            center: myLatLng,
            disableDefaultUI: true,
            disableDoubleClickZoom: false,
            draggable: true,
            mapTypeId: google.maps.MapTypeId.ROADMAP
        };
        var map = new google.maps.Map(document.
        getElementById("map_canvas"),
        myOptions);
        //输出 JavaScript 语句添加叠加图
        .....
    }
    //添加单个叠加图
    function addMarker(v_map, v_lat, v_lng, v_title) {
        var marker = new google.maps.Marker( {
            position: new google.maps.LatLng (v_lat/
            1000000, v_lng/1000000),
            title: v_title
        });
        //将添加叠加图到地图中
        marker.setMap(v_map);
    }
</script>
</head>
<body onload="initialize()">
    <div id="map_canvas"></div>
</body>
</html>
```

```
var map = new google.maps.Map (document.
getElementById("map_canvas"),
myOptions);
//输出 JavaScript 语句添加叠加图
.....
}
//添加单个叠加图
function addMarker(v_map, v_lat, v_lng, v_title) {
    var marker = new google.maps.Marker( {
        position: new google.maps.LatLng (v_lat/
        1000000, v_lng/1000000),
        title: v_title
    });
    //将添加叠加图到地图中
    marker.setMap(v_map);
}
</script>
</head>
<body onload="initialize()">
    <div id="map_canvas"></div>
</body>
</html>
```

代码 9 中，使用了基本的地图 API 功能，即：显示 Web 地图和添加叠加图。其中 Map 对象用于构建 Web 地图，其需要一个网页组件 DIV 来承载地图，另外就是用于设置地图选项的属性集（myOptions）。

地图选项包含很多内容，常见的有：缩放比例、中心点坐标、地图类型以及操作许可等，详细的内容可以参考谷歌地图 JavaScript API。

在代码 9 中，服务端只需要将经纬度坐标串按照 JavaScript 函数输出成 JavaScript 语句块即可。

### 3.3.3 以路径的方式展示轨迹

代码 10 是 JSP 后端生成轨迹坐标点的 JavaScript 语句的主要代码。

```
//代码 10 生成地图 JavaScript 语句
//文件名:query_markers.jsp
<%
    //连接 SQLite 数据库
    final String filePath = baseRealPath+"track_locations.db";
    Connection conn = SQLite3Util.getInstance().
    getConnection(filePath);
    if(conn==null) {
        return;
    }
    //纬度和经度偏差
    final String lat_diff = "-2000";
    final String lon_diff = "+6000";
```





## COMPUTER SECURITY AND MAINTENANCE

```
//使用经纬度列生成 JavaScript 语句
ArrayList<String> items = SQLite3Util.getInstance().
getArrayList(conn,
    "SELECT ('addMarker(map,
        '||(t.lat*+lat_diff+)'||','||(t.lon*+lon_diff+)'||','||t.
        '_time'||'\n');') item "+
    "FROM 'tab_locations' t, 'tab_track_list' t2 "+
    "WHERE (t.track_id = t2._id' AND t2._id' = "+
    trackId+') order by t._id;");
if(items==null) {
    return;
}
//关闭数据库连接
try {
    conn.close();
} catch (SQLException e) {
    e.printStackTrace();
}
//输出 JavaScript 语句块
for(int i = 0; i < items.size(); ++i) {
    out.println("\t\t"+items.get(i));
}
```

(上接第 72 页)

```
new SolidBrush (Color.DarkSlateGray), new PointF (100,
150));
}
//重新绘制贪吃蛇
public void DrawSnakeMove(Point mp)
{ bool tem_bool = false;//是否清除移动后的蛇身
  SnakeMode.List.Insert(0, mp);//插入新的蛇头的位置
  //记录蛇尾的位置
  Point tem_point = ((Point)SnakeMode.List
[SnakeMode.List.Count - 1]);SnakeMode.List.RemoveAt(Sna
keMode.List.Count - 1);//移除蛇的尾部
  //使骨节向前移动一位
  for (int i = 0; i < SnakeMode.List.Count - 1; i++)
  { if (tem_point == ((Point)SnakeMode.List[i]))
    tem_bool = true;
  }
  if (! tem_bool)//清除贪吃蛇移动前的蛇尾部份
  g.FillRectangle (SolidD, tem_point.X + 1, tem_point.Y +
1, SnakeMode.Node - 1, SnakeMode.Node - 1);
  //重新绘制蛇头
  g.FillRectangle (SolidB, ((Point)SnakeMode.List [0]).X
+ 1, ((Point)SnakeMode.List[0]).Y + 1, SnakeMode.Node - 1,
SnakeMode.Node - 1);
}
//清空游戏背景
public void BackgroundClear()
{ if (Mycontrol != null)//如要已载入背景控件
  { Rectangle rect = new Rectangle(0, 0, Mycontrol.
```

```
}
%>
```

代码 10 中, 从数据库中获取指定日期内的经纬度坐标点并生成地图 JavaScript 语句, 然后将这些语句输出到前端页面。

其中, 坐标点按照时间点的先后进行排序, 叠加图的文字内容即为时间点。其页面效果如图 1 所示。

#### 4 结语

在内容上凸显了 3 个方面的要点, 第一, 使用手机基站实现了低成本实时定位 (无需 3G 网络和 GPS 模块); 第二, 是 Web 地图应用的集成; 第三, 方案完整, 各环节的交互多样 (手机端是通过 SD 卡存储和 Wi-Fi 传输; 桌面端使用 SQLite 数据库和 Web Service 访问; 服务端使用的是 BS 架构和 Web 地图 API 交互)。

该案例离一个真正的商业项目的要求还是比较远, 例如: 如何实现实时监控、定位精度还有待提高、经纬度的查询对外部服务的依赖过强等。鉴于此, 笔者希望抛砖引玉, 通过此文而得到更多读者的应用经验, 从而不断提高本案例的应用水平。

(收稿日期: 2012-08-15)

```
Width, Mycontrol.Height); g.FillRectangle (new SolidBrush
(Mycontrol.BackColor), rect);//用背景色填充背景
```

```
}
}
// 绘制游戏场景
public void DrawBackground()
{ for (int i = 0; i <= Mycontrol.Width / SnakeMode.
Node; i++)//绘制单元格的纵向线
{ g.DrawLine(new Pen(Color.Black, 1), new Point(i
* SnakeMode.Node, 0),
new Point(i * SnakeMode.Node, Mycontrol.Height));
}
for (int i = 0; i <= Mycontrol.Height / SnakeMode.
Node; i++)//绘制单元格的横向线
{ g.DrawLine(new Pen(Color.Black, 1), new Point(0,
i * SnakeMode.Node),
new Point(Mycontrol.Width, i * SnakeMode.Node));
}
}
}
```

#### 4 结语

利用 C# 开发贪吃蛇游戏, 借鉴了 MVC 的设计模式, 通过建立模型类、游戏规则类和画图类, 使程序流程更加清晰, 并且程序具有强内聚、松耦合特性, 使功能扩展更加方便。该方法对于开发类似的 GDI+ 游戏有很好的借鉴作用。

(收稿日期: 2012-09-07)



# Visual Assist X 与 VBA 结合实现在 Word 里排版代码

刘 烽

**摘 要:** 针对在 Word 里排版代码质量不高的现实困难, 利用 Visual Assist X VS 插件与 VBA 相结合的方法, 较好地解决了这个问题。通过此方法整理后的代码, 简洁醒目, 便于阅读, 大大提高了稿件的排版质量。

**关键词:** Visual Assist 工具; VBA 编程; 代码排版

## 1 引言

随着微软 Office 在国内的深入普及, 多数国内电子科技类杂志社通常都要求 Word 稿件, 并对稿件的文字、图表、公式都有相应的格式要求。笔者在投稿中发现: 大多数杂志社没有对稿件中的代码排版作出规范, 导致部分稿件中代码排版不美观, 给读者在阅读代码时候带来诸多不便。笔者个人认为好的代码排版应当具备以下几个特点: (1) 良好的缩进, 语法关键字高亮。(2) 代码附有行号。(3) 整个代码排版风格统一, 在文中布局美观。但是在 Word 中, 针对上述特点对代码进行漂亮的排版还是比较困难的。经过一段时间的探索, 笔者发现利用 Visual Studio+Visual Assist X+VBA 的结合便能够对代码进行美观的排版, 从而大大提高稿件的排版质量。

众所周知, Visual Studio 是微软公司推出的开发环境, 集成了 C#、C/C++、J# 等多种语言, 是 Windows 平台应用程序开发最流行的 IDE 环境, 相关的版本有 Visual Studio 2003、Visual Studio 2005、Visual Studio 2008 以及最新的 Visual Studio 2012。Visual Assist X 是一款非常好的 Visual Studio 插件, 几乎支持所有 Visual Studio 版本, 支持 C/C++, C#, ASP, Visual Basic、Java 及 HTML 等语言, 能自动识别各种关键字、系统函数、成员变量、自动给出输入提示、自动更正大小写错误、自动标示错误等, 能大大提高开发过程自动化和开发效率。

## 2 设计思路

针对上述代码排版的几个特点, 分别介绍如何实现源代码在 Word 中的漂亮排版。

(1) 实现代码在 Word 中良好的缩进和语法关键字高亮。

对于在 Visual Studio 2003 以上版本开发的代码, 用户直接将 IDE 环境中需要拷贝的代码选中, 按组合键 ALT+F8 调整好缩进后, 直接复制粘贴至 Word 文档中即可实现代码良好的缩进和代码高亮, 操作较为简单。而对于习惯于 VC6.0 的用户, 如果直接将代码粘贴至 Word 中, 字体和格式都将发生变

化, 代码没有高亮效果, 结合 Visual Assist X 就可完成代码的高亮排版。下载 Visual Assist X 安装程序并安装, 打开 VC6.0 后, 会生成 Visual Assist X 工具条 (如图 1 所示)。



图 1 Visual Assist X 工具条

点击工具条中的按钮, 弹出 Visual Assist X Options 对话框, 勾选 “Copy RTF to the clipboard” 选项 (如图 2 所示), 这样, 代码粘贴至 Word 中便具有语法高亮效果, 如图 3 所示。



图 2 Visual Assist X Options 对话框

```
Test.cpp: Defines the entry point for the console application.
...
#include <stdafx.h>
#include <math.h>
#include <stdlib.h>
#include <iostream>
using namespace std;
inline double RandFloat(int x, int y) { return (rand() / (RAND_MAX + 1.0)) * abs(y - x) + x; }
int main(int argc, char* args[])
{
    double d, m;
    d = RandFloat(0.4);
    m = RandFloat(1.5);
    cout << d << " " << (int)d << endl;
    cout << m << " " << (int)m << endl;
    cout << d << " " << float(d) << endl;
    cout << d << " " << ceil(d) << endl;
    printf("Hello World!\n");
    return 0;
}
```

图 3 代码从 VC6.0 拷贝至 Word 中语法高亮效果



## COMPUTER SECURITY AND MAINTENANCE

(2) 实现在代码中添加行号。

利用 Word VBA 编程的强大功能, 在 Word2007 中, 按组合键 ALT+F8, 弹出“宏”对话框, 在“宏名”文本框中输入“AddCodeLine”, 然后单击“创建”按钮, 进入 Word VBA 编辑环境。这里笔者较喜欢的一种代码排版风格是: 行号为斜体, 蓝色标注, 为了在文章中突出代码环境且便于阅读, 代码着背景色, 同时相邻行之间背景色不同。完成此功能的 VBA 代码如下:

```
#001 Sub AddCodeLine()
#002 '统计代码行数
#003 Dim ParagraphsCount As Integer
#004 ParagraphsCount = Selection.Paragraphs.Count
#005 '将代码转化为表格,并设置表格风格
#006 WordBasic.TextToTable ConvertFrom:=0, NumColumns:=1, NumRows:=ParagraphsCount, _
#007 InitialColWidth:=wdAutoPosition, Format:=0, Apply:=1184, AutoFit:=0, _
#008 SetDefault:=0, Word8:=0, Style:="网格型"
#009 Selection.Tables(1).Style = "中等深浅网格 1 - 强调文字颜色 1"
#010 With Selection.Tables(1)
#011 .Borders(wdBorderLeft).LineStyle = wdLineStyleNone
#012 .Borders(wdBorderRight).LineStyle = wdLineStyleNone
#013 .Borders(wdBorderTop).LineStyle = wdLineStyleNone
#014 .Borders(wdBorderBottom).LineStyle = wdLineStyleNone
#015 .Borders(wdBorderHorizontal).LineStyle = wdLineStyleNone
#016 .Borders(wdBorderDiagonalDown).LineStyle = wdLineStyleNone
#017 .Borders(wdBorderDiagonalUp).LineStyle = wdLineStyleNone
#018 .Borders.Shadow = False
#019 End With
#020 With Options
#021 .DefaultBorderLineStyle = wdLineStyleSingle
#022 .DefaultBorderLineWidth = wdLineWidth100pt
#023 .DefaultBorderColor = -738132033
#024 End With
#025 '插入一列,用于将显示行号
#026 Selection.InsertColumns
#027 Selection.Tables(1).Columns(1).SetWidth ColumnWidth:=60, RulerStyle:= _
#028 wdAdjustFirstColumn
#029 Dim i As Integer
#030 Dim CodeLine As String
#031 For i = 1 To ParagraphsCount
#032 '通常在稿件中附加的代码不宜过长,这里设置最多支持行代码。
#033 CodeLine = "#" + Format(i, "000")
```

```
#034 Selection.Tables(1).Cell(i, 1).Range.InsertAfter(CodeLine)
#035 Next i
#036 '设置行号格式
#037 Selection.Tables(1).Columns(1).Select()
#038 Selection.Font.Color = 12611584
#039 Selection.Font.Italic = wdToggle
#040 Selection.Font.Size = 9
#041 Selection.Font.Bold = wdToggle
#042 Selection.Cells.VerticalAlignment = wdCellAlignVerticalCenter
#043 Selection.Tables(1).AutoFitBehavior wdAutoFitContent
#044End Sub
```

上述示例代码的排版正是利用 AddCodeLine () 实现添加行号和背景着色的效果的。运行这段程序很简单, 具体方法为: 选中所有需要格式化排版的代码, 然后在“宏”对话框中选择“宏名”为 AddCodeLine 的宏, 单击“运行”。即可将代码进行如上效果的排版。

### 3 结语

利用 VS+Visual Assist X 和 VBA 的结合, 可以很方便地在 Word 里排版高质量的代码文档, 其效果几乎可与专业的排版软件 LaTeX 相媲美, 而且方法简单快速, 与此同时, 只要将文中的代码稍加改进, 可以开发出功能更加强大的 Word 代码排版小插件, 美化稿件质量。

(收稿日期: 2012-11-22)

#### 苹果或在中国推低价 iPhone

昨日, 有分析师预计, 苹果可能会于今年夏天在中国市场上推出 iPhone mini, 而且裸机售价预计为 330 美元 (大约 2000 元人民币)。有手机业资深人士昨日表示, 苹果盈利能力开始下降, 在中国市场, 三星更是在市场份额上把苹果甩得很远, 因此苹果推出低价 iPhone 应对当前困境是有可能的。

摩根斯坦利分析师凯蒂·胡伯特昨日发布报告表示, 苹果可能会于今年夏天在中国市场上推出 iPhone mini, 其裸机售价预计为 330 美元, 而此前有关苹果将会推出低价 iPhone 的传闻不断出现, 甚至有多名分析师也预计, 苹果今年将会推出低价 iPhone。

战国策首席分析师杨群昨日表示, 从 2012 年下半年开始, 手机市场的“明星”产品发生了变化, 此前很受欢迎的苹果 iPhone 让位于三星的“盖世”系列。由于 iPhone 只有一款产品, 且价格一直居高不下, 等于自己放弃了中低端市场。



## Delphi 开发中的复用技巧

张扬嵩

**摘 要:** 在软件开发的过程中, 软件复用是提高软件开发效率和质量的一个重要手段。软件复用的方法和技巧很多, 介绍一种在 Delphi 数据库应用开发中有效复用的方法和技巧, 通过该方法能大大提高 Delphi 数据库应用的开发效率。

**关键词:** 设计模式; 继承; 抽象类; 软件复用

### 1 引言

Delphi 是一个在 Windows 操作系统平台上比较流行的开发工具、也是一个比较容易使用的快速工发工具, 并且 Delphi 对数据库有比较好的支持, 通过 Delphi 的 ADO 或 BDE, 比较容易和方便访问多种数据库。因为现在很多应用软件的开发都有应用到数据库的支持, 所以在很多企业级应用的开发中常选用 Delphi 作为开发工具。Delphi 也是一种面向对象的开发语言, 可以充分利用面向对象的开发技术和技巧来提高开发效率和质量。封装和继承是面向对象软件开发中最基本的支持。下面将介绍笔者在实际的数据库应用开发中, 应用模板的设计模式, 通过继承父类的方法来达到对数据库表操作窗体复用的技巧。

### 2 应用场景

对数据库表的操作一般包含增加、删除、修改和查询数据库表, 这个处理逻辑都是一样的, 可以封装在父类实现, 在子类只要负责定义要操作的表名, 要操作的字段, 要操作的条件这些设置信息就可以了, 这样子类的就没有逻辑的处理, 所有的逻辑实现都在父类实现。在这里, 为了说明简单和方便, 数据库用要本地 Access 的数据库, 实际应用中要根据具体应用选择 Delphi 支持可访问的数据库。

### 3 实现实例

新建一个 Delphi 工程, 加入一个数据库访问模块 (Data Module), 在这个窗体上加上一个 ADOConnection 组件对象, 并进行设置使其正确访问所要操作的数据库。这个例子里是访问本地 Access 数据库, 关键设置:

Provider=Microsoft.Jet.OLEDB.4.0;User ID=Admin;Data Source=.\dbdev.mdb;

Mode=Share Deny None;Extended Properties="" ;Jet OLEDB: System database="" ;这个属性就可以访问本地 Access 数据库了。正确访问数据库后, 再新建一个窗体模块, 这个窗体模块

的功能是实现数据库表的增删改查的操作功能。这个窗体设置很简单, 放一个 TDBGrid 用于展现数据, 放 3 个 TADOQuery 对象用于操作数据库表, 中间放一个 TPanel 用于子类继承时放要操作的数据库表字段信息, 最底下就是操作的按钮, 这些按钮事件全部放在子类处理, 为了这个类处理的统一, 在这里, 所有放在 TPanel 上的子控件的命名要求是控件类型加下划线加字段名, 如: ed\_code 这样; 代码如下:

```
unit uAbsDeal;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  Grids, DBGrids, Db, ADODB, StdCtrls, Buttons, ExtCtrls,
  hashes, udb;
type
  TfrmAbsDeal = class(TForm)
    Panel1: TPanel;
    pnl_view: TPanel;
    pnl4Edits: TPanel;
    pnl_ops: TPanel;
    BitBtnAdd: TBitBtn;
    BitBtnUpd: TBitBtn;
    BitBtnDel: TBitBtn;
    BitBtnQry: TBitBtn;
    BitBtnReset: TBitBtn;
    ADODml: TADOQuery;
    ADOQry: TADOQuery;
    dsMid: TDataSource;
    DBGridCfgData: TDBGrid;
    bitBtn_Close: TBitBtn;
    ADOQryRe: TADOQuery;
    procedure BitBtnResetClick(Sender: TObject);
    procedure ADOQryAfterScroll(DataSet: TDataSet);
    procedure bitBtn_CloseClick(Sender: TObject);
    procedure FormClose (Sender: TObject; var Action:
    TCloseAction);
```





## COMPUTER SECURITY AND MAINTENANCE

```

procedure FormShow(Sender: TObject);
    procedure FormCanResize (Sender: TObject; var
NewWidth,
        NewHeight: Integer; var Resize: Boolean);
private
    { Private declarations }
public
    { Public declarations }
    tabName, //要对哪张表操作
    opType, //操作类型
    qryCond: String; //条件
    flds, //要对哪张表的字段操作
    keyflds: THash; //要对哪张表的关键字段操作
    lbls: TObjectHash;
    function DealUpd(): String;
    function DealAdd(): String;
    function DealDel(): String;
    function DealQry(): String;
    function CheckVal(var chkMsg: String): Boolean;
    function CheckExist(var msgInfo: String): Boolean;
end;
var
    frmAbsDeal: TfrmAbsDeal;
implementation
{$R *.DFM}
{ TfrmAbsCfg }
function TfrmAbsDeal.CheckVal (var chkMsg: String):
Boolean;
var
    idx,cnt: Integer;
    function GetLblCaption(lblName: String): String;
    var
        idx2,cnt2: Integer;
    begin
        Result := '';
        cnt2 := pnl4Edits.ControlCount - 1;
        for idx2 := 0 to cnt2 do
            begin
                if pnl4Edits.Controls[idx2] is TLabel then
                    begin
                        if (UpperCase(TLabel(pnl4Edits.Controls[idx2]).Name) =
UpperCase(lblName)) then
                            begin
                                Result := TLabel(pnl4Edits.Controls[idx2]).Caption;
                                Exit;
                            end;
                        end;
                    end;
                end;
            end;
        end;//GetLblCaption
begin
    Result := True;
    chkMsg := '';

```

```

cnt := pnl4Edits.ControlCount - 1;
for idx := 0 to cnt do
begin
if pnl4Edits.Controls[idx] is TEdit then
begin
if (TEdit(pnl4Edits.Controls[idx]).Tag >= 11)
and (Trim(TEdit(pnl4Edits.Controls[idx]).Text) = '') then
begin
chkMsg := GetLblCaption('lbl_' + Copy(TEdit(pnl4Edits
Controls [idx]).Name,4,length (TEdit (pnl4Edits.Controls[idx]).
Name)))
+ ' 不能为空,请输入 ';
Result := False;
Exit;
end;
end else if pnl4Edits.Controls[idx] is TComboBox then
begin
if (TComboBox(pnl4Edits.Controls[idx]).Tag >= 11)
and (Trim (TComboBox (pnl4Edits.Controls [idx]).Text) =
'') then
begin
chkMsg := GetLblCaption ('lbl_' + Copy (TComboBox
(pnl4Edits.Controls [idx]).Name,5,length (TComboBox
(pnl4Edits.Controls[idx]).Name)))
+ ' 不能为空,请输入 ';
Result := False;
Exit;
end;
end;
end;
end;
function TfrmAbsDeal.DealAdd: String;
var
idx,cnt: Integer;
strsql,strsql2,strvals,strMsg: String;
fldNames: TStringList;
begin
if CheckExist(strMsg) then
begin
ShowMessage(strMsg+' 数据已经存在,不能再增加. ');
Exit;
end;
strsql := 'insert into ' + tabName + '(';
strvals := ' values(';
fldNames := keyflds.Keys;
cnt := fldNames.Count - 1;
for idx := 0 to cnt do
begin
strsql := strsql + fldNames[idx] + ',';
strvals := strvals + "'" + keyflds.GetString (fldNames[idx])
+ "'";
end;

```

```

fldNames := flds.Keys;
cnt := fldNames.Count - 1;
for idx := 0 to cnt do
begin
    strsql := strsql + fldNames[idx] + ',';
strvals := strvals + '""' + flds.GetString(fldNames[idx]) + '""';
end;
strsql2 := copy(strsql,0,length(strsql)-1) + ')'
    + copy(strvals,0,length(strvals)-1) + ')';
ADODml.Close;
ADODml.SQL.Text := strsql2;
ADODml.ExecSQL;
MsgDlg('新增成功!',mtinformation,[mbOK],0);
end;
function TfrmAbsDeal.DealDel: String;
var
    idx,cnt: Integer;
    strsql,strsql2: String;
    fldNames: TStringList;
begin
    if MsgDlg('确定删除该条记录?',mtConfirmation,
[mbYes,mbNo],0)<>mryes then
        exit;
    strsql := 'delete from ' + tabName;
    fldNames := keyflds.Keys;
    cnt := fldNames.Count - 1;
    for idx := 0 to cnt do
    begin
        strsql := strsql + ' where ' + fldNames [idx] + '=' +
keyflds.GetString(fldNames[idx]) + '""';
    end;
    strsql2 := copy(strsql,0,length(strsql)-1);
    ADODml.Close;
    ADODml.SQL.Text := strsql2;
    ADODml.ExecSQL;
    MsgDlg('删除成功!',mtinformation,[mbOK],0);
end;
function TfrmAbsDeal.DealQry: String;
var
    idx,cnt: Integer;
    strCond,strsql,strsql2: String;
    fldNames: TStringList;
begin
    cnt := pnl4Edits.ControlCount - 1;
    if opType = 'qry' then
    begin
        strCond := ' where 1=1';
        for idx := 0 to cnt do
        begin
            if pnl4Edits.Controls[idx] is TEdit then
            begin
                if (TEdit(pnl4Edits.Controls[idx]).Tag = 12)

```

```

and (Trim(TEdit(pnl4Edits.Controls[idx]).Text) <> '') then
    begin
        strCond := strCond + ' and '
            + Copy(TEdit(pnl4Edits.Controls[idx]).name,4,length
(TEdit(pnl4Edits.Controls[idx]).name))
            + '=' + Trim(TEdit(pnl4Edits.Controls[idx]).Text) + '""';
    end;
end else if pnl4Edits.Controls[idx] is TComboBox then
    begin
        if (TComboBox(pnl4Edits.Controls[idx]).Tag = 12)
            and (Trim (TComboBox (pnl4Edits.Controls [idx]).Text)
<> '') then
            begin
                strCond := strCond + ' and '
                    + Copy (TComboBox (pnl4Edits.Controls [idx]).name,
4,length(TComboBox(pnl4Edits.Controls[idx]).name))
                    + '=' + TComboBox(pnl4Edits.Controls[idx]).text + '""';
            end;
        end
    else if pnl4Edits.Controls[idx] is TCheckBox then
        begin
            if (TCheckBox(pnl4Edits.Controls[idx]).Tag = 12)
            and (TCheckBox(pnl4Edits.Controls[idx]).Checked) then
                begin
                    strCond := strCond + ' and '
                        + Copy(TCheckBox(pnl4Edits.Controls[idx]).name,5,
length(TCheckBox(pnl4Edits.Controls[idx]).name))
                        + '=' + '1' + '""';
                end;
            end;
        end;
    end else
        strCond := qryCond;
    strsql := 'select ' ;
    fldNames := keyflds.Keys;
    cnt := fldNames.Count - 1;
    for idx := 0 to cnt do
    begin
        strsql := strsql + fldNames[idx] + ',';
    end;
    fldNames := flds.Keys;
    cnt := fldNames.Count - 1;
    for idx := 0 to cnt do
    begin
        strsql := strsql + fldNames[idx] + ',';
    end;
    strsql2 := copy(strsql,0,length(strsql)-1);
    strsql2 := strsql2 + ' from ' + tabName + strCond;
    ADOQry.Close;
    ADOQry.SQL.Text := strsql2;
    ADOQry.Open;
    qryCond := strCond;

```





# COMPUTER SECURITY AND MAINTENANCE

```

end;
function TfrmAbsDeal.DealUpd: String;
var
  idx,cnt: Integer;
  strsql,strsql2: String;
  fldNames: TStringList;
begin
  strsql := 'update ' + tabName + ' set ';
  fldNames := flds.Keys;
  cnt := fldNames.Count - 1;
  for idx := 0 to cnt do
  begin
    strsql := strsql + fldNames [idx] + '=' + flds.GetString
(fldNames[idx]) + ',';
  end;
  strsql2 := copy(strsql,0,length(strsql)-1);
  fldNames := keyflds.Keys;
  cnt := fldNames.Count - 1;
  for idx := 0 to cnt do
  begin
    strsql := strsql2 + ' where ' + fldNames [idx] + '=' +
keyflds.GetString(fldNames[idx]) + ',';
  end;
  strsql2 := copy(strsql,0,length(strsql)-1);
  ADODml.Close;
  ADODml.SQL.Text := strsql2;
  ADODml.execSQL;
  MessageDlg('修改成功!',mtInformation,[mbOK],0);
end;
procedure TfrmAbsDeal.BitBtnResetClick(Sender: TObject);
var
  idx,cnt: Integer;
begin
  cnt := pnl4Edits.ControlCount - 1;
  for idx := 0 to cnt do
  begin
    if pnl4Edits.Controls[idx] is TEdit then
      TEdit(pnl4Edits.Controls[idx]).Text := ''
    else if pnl4Edits.Controls[idx] is TComboBox then
      TComboBox(pnl4Edits.Controls[idx]).ItemIndex := 0
    else if pnl4Edits.Controls[idx] is TCheckBox then
      TCheckBox(pnl4Edits.Controls[idx]).Checked := False;
  end;
end;
procedure TfrmAbsDeal.ADOQryAfterScroll(DataSet:
TDataSet);
var
  idx,cnt,v_ItemIndex: Integer;
  fldname,v_cmbvalue,v_code,v_name,v_debugname: String;
begin
  cnt := pnl4Edits.ControlCount - 1;
  for idx := 0 to cnt do

```

```

begin
  if (pnl4Edits.Controls[idx] is TEdit) then
  begin
    fldname := Copy (TEdit (pnl4Edits.Controls [idx]).Name,4,
length(TEdit(pnl4Edits.Controls[idx]).Name));
    TEdit (pnl4Edits.Controls [idx]).Text := ADOQry.
fieldbyname(fldname).AsString;
  end else if pnl4Edits.Controls[idx] is TComboBox then
  begin
    fldname := Copy (TComboBox (pnl4Edits.Controls[idx]).
Name,4,length(TComboBox(pnl4Edits.Controls[idx]).Name));
    try
      v_cmbvalue:=ADOQry.fieldbyname(fldname).AsString;
      TComboBox (pnl4Edits.Controls [idx]).ItemIndex :=
TComboBox(pnl4Edits.Controls[idx]).Items.IndexOf(v_cmbva
lue);
    except
    end;
  end
  else if pnl4Edits.Controls[idx] is TCheckBox then
  begin
    fldname := Copy (TCheckBox (pnl4Edits.Controls [idx]).
Name,4,length(TCheckBox(pnl4Edits.Controls[idx]).Name));
    if ADOQry.fieldbyname(fldname).AsString = '1' then
      TCheckBox(pnl4Edits.Controls[idx]).Checked := True
    else
      TCheckBox(pnl4Edits.Controls[idx]).Checked := False;
  end;
end;
end;
procedure TfrmAbsDeal.bitBtn_CloseClick(Sender: TObject);
begin
  Close();
end;
function TfrmAbsDeal.CheckExist (var msgInfo: String):
Boolean;
var
  strsql: String;
  idx, cnt: Integer;
  fldNames: TStringList;
begin
  fldNames := keyflds.Keys;
  strsql := 'select 1 from ' + tabName + ' where 1=1 ';
  cnt := keyflds.Count - 1;
  for idx := 0 to cnt do
  begin
    strsql := strsql + ' and ' + fldNames[idx] + '=' +
+ keyflds.GetString(fldNames[idx]) + ',';
    msgInfo := fldNames [idx] + ':' + keyflds.GetString
(fldNames[idx]) + ' ';
  end;
  if cnt < 0 then

```



```

    strsql := strsql + ' and 1<>1 ' ;
ADODml.Close;
ADODml.SQL.Text := strsql;
ADODml.Open;
if ADODml.IsEmpty then
    Result := False
else
    Result := True;
end;
procedure TfrmAbsDeal.FormClose (Sender: TObject; var
Action: TCloseAction);
begin
    if nil <> flds then
    begin
        flds.Free;
        flds := nil;
    end;
    if nil <> keyflds then
    begin
        keyflds.Free;
        keyflds := nil;
    end;
    Action:=caFree;
    frmAbsDeal:=nil;
end;
procedure TfrmAbsDeal.FormShow(Sender: TObject);
begin
    DBGridCfgData.SetFocus;
end;
procedure TfrmAbsDeal.FormCanResize(Sender: TObject; var
NewWidth,
    NewHeight: Integer; var Resize: Boolean);
begin
    DBGridCfgData.SetFocus;
end;
end.
```

这个类里的 DealAdd、DealDel、DealUpd、DealQry 方法封装了对数据库表的增、删、改、查操作，当然要操作的表和字段都是未知的，必须要有子类来实现，先编译通过后，再新建一个窗体模块，并且在 New Items 上选择当前工程页里的 frmAbsDeal 作为父类继承，这样新建的窗体就从 frmAbsDeal 窗体继承了功能和框架；在这个新建的窗体上放置一些编辑框并进行正确的设置，设置 TDBGrid 的字段信息，然后在按钮的事件中做些设置就行了，在这个类里都没有逻辑处理，具体代码如下：

```

unit uInfoDeal;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics,
    Controls, Forms, Dialogs,
    Db, ADODB, StdCtrls, Buttons, Grids, DBGrids, ExtCtrls,
```

```

hashes, uAbsDeal;
type
    TfrmCfgInfo = class(TfrmAbsDeal)
        ed_code: TEdit; //命名规则:控件类型加下划线加字段名
        Ed_name: TEdit;
        lbl_Name: TLabel;
        ed_type: TEdit;
        lbl_code: TLabel;
        lbl_type: TLabel;
        procedure FormCreate(Sender: TObject);
        procedure BitBtnDelClick(Sender: TObject);
        procedure BitBtnAddClick(Sender: TObject);
        procedure BitBtnQryClick(Sender: TObject);
        procedure BitBtnUpdClick(Sender: TObject);
        procedure FormClose (Sender: TObject; var Action:
TCloseAction);
        procedure bitBtn_CloseClick(Sender: TObject);
        procedure BitBtnResetClick(Sender: TObject);
    end;
var
    frmCfgInfo: TfrmCfgInfo;
implementation
{$R *.DFM}
procedure TfrmCfgInfo.FormCreate(Sender: TObject);
begin//初始化设置
    opType := 'qry';
    tabName := 'tinfo';
    flds := THash.Create;
    flds.SetString('type','');
    flds.SetString('name','');
    keyflds := THash.Create;
    keyflds.SetString('code','');
    inherited DealQry;
end;
procedure TfrmCfgInfo.BitBtnDelClick(Sender: TObject);
begin
    opType := 'del';
    keyflds.SetString('code',Ed_code.Text);
    inherited DealDel;
    inherited DealQry;
end;
procedure TfrmCfgInfo.BitBtnAddClick(Sender: TObject);
var
    strmsg: string;
begin
    opType := 'add';
    flds.SetString('type',Ed_type.Text);
    flds.SetString('name',Ed_Name.Text);
    keyflds.SetString('code',Ed_code.Text);
    if not CheckVal(strmsg) then
    begin
        ShowMessage(strmsg);
```





## COMPUTER SECURITY AND MAINTENANCE

```

Exit;
end;
inherited DealAdd;
inherited DealQry;
end;
procedure TfrmCfglInfo.BitBtnQryClick(Sender: TObject);
begin
    opType := 'qry';
    flds.SetString('type',Ed_Type.Text);
    flds.SetString('name',Ed_Name.Text);
    flds.SetString('code',Ed_code.Text);
    inherited DealQry;
end;
procedure TfrmCfglInfo.BitBtnUpdClick(Sender: TObject);
begin
    opType := 'upd';
    flds.SetString('type',Ed_Type.Text);
    flds.SetString('name',Ed_Name.Text);
    flds.SetString('code',Ed_code.Text);
    flds.SetString('code',Ed_code.Text);
    inherited DealUpd;
    inherited DealQry;
end;
procedure TfrmCfglInfo.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
    Action:=caFree;

```

```

frmCfglInfo:=nil;
inherited;
end;
procedure TfrmCfglInfo.BitBtn_CloseClick(Sender: TObject);
begin
    inherited;
end;
procedure TfrmCfglInfo.BitBtnResetClick(Sender: TObject);
begin
    inherited;
end;
end.

```

这个子类里，主要就是要定义操作的表名、字段名和关键字段名，很简单和快捷地就实现了对一个表的所有操作，这主要是复用了父类的功能。

## 4 结语

通过在父类窗体模块里实现对数据库表所有的逻辑处理，只要是对数据库表操作的窗体模块都可以继承，继承后的子类窗体只要定义操作的表名、字段名和关键字段名就可以实现对一个表的所有操作，最大限度地实现软件复用，大大提高了开发效率。希望能起到抛砖引玉的作用，让大家发现更多在软件开发中提高软件复用和开发效率。

(收稿日期：2012-10-09)

## 湖南龙畅集团携手用友信息化腾飞

2013 年始，用友株洲分公司交出了第一份答卷！2013 年 1 月 28 日，在湖南龙畅生态农业集团有限公司会议室，用友软件株洲分公司与湖南龙畅生态农业集团打成合作，顺利签订战略合作协议！

本次签约用友软件股份有限公司副董事长郭新平、湖南龙畅投资集团有限公司董事长唐定龙、省经信委领导、市经信委领导及当地政府领导班子出席了本次会议。

湖南龙畅投资集团有限公司始创于 1992 年，是一家集现代物流、环保能源、生态农业、商业地产、汽车贸易等多个产业于一体的综合型民营企业。集团下辖湖南龙畅集团洁净煤有限公司、物流有限公司、生态农业有限公司、房地产有限公司、金源汽贸共五家控股子公司。其中物流公司为国家 AAAA 级综合型物流企业、国家税收总局物流税收试点单位、湖南省物流与采购联合会副会长单位，在省内排名前列；湖南龙畅洁净煤有限公司煤炭洗选工艺全省领先，规模列全省前三甲；生态农业、商业地产瞄准行业前沿，正处于蓬勃发展期。

本次签约的龙畅生态农业集团致力现代农业、生态农业、文化农业、效益农业，遵循自然规律的生态文明理念；以“食和食文化”为主，“游和乐”为辅的产业战略布局；围绕“健康、养生、尊贵”的生活理念，重点打造“四大功能实体”——农业迪斯尼乐园、原生态生产农场、原生态加工工厂、龙畅金谷生态广场。公司板块客户预计打造中国领先农业品牌，定位高端人群，打造线上线下运营体系。在十五五末在全国开设农业迪斯尼乐园、生产农场和生态广场千余家，产值过万亿，会员超 1 亿；并于 2015 年实现上市。

作为客户信赖的长期合作伙伴，用友公司将凭借多年累积的信息化建设经验、专业完善的行业解决方案及成熟的产品，从客户的需求出发，以专业的能力、诚信负责的态度，不断创造先进产品长期优质服务客户，帮助客户持续成功！

相信在不久的将来，用友株洲分公司将与龙畅集团一起携手共同谱写更多的发展新篇章！



## Q 如何用 OpenLu 解方程

**A** 解方程(组)是工程计算、数值计算中很普遍的现象。人们用计算机求解,大大提高了工作效率。

解方程(组)有符号求解和数值求解两种。工程中遇到的方程大多没有符号解,而只有数值解,故数值求解是更广泛应用的方法。下文仅讨论方程(组)的数值求解。

目前,Matlab、Mathematica、Maple等数学软件都具有强大的解方程能力,但普遍存在的问题是:

- (1) 需要初值,这是令人十分头痛的问题。
- (2) 在需要获得方程(组)全部解时,需要复杂的编程实现。

推荐使用 OpenLu 求解方程(组)。OpenLu 是国产软件,绿色免安装,体积小巧而又不失功能强大,在解方程方面尤其出色。OpenLu 在解方程时,不需要初值,而且容易获得方程的全部解,在方程(组)的软件求解方面前进了一大步。

OpenLu 使用 Lu 脚本作为编程语言。Lu 是一种易于扩展的轻量级嵌入式脚本,代码简洁、运行效率高,具有丰富的运算符和数据类型,非常适合于数值计算。

方程(组)的求解,难易程度差别较大。在 OpenLu 中,普通的方程(组)可借助 LuMath 库中的拟牛顿法 `netn` 和对分法 `btFindRoot` 求解,难度大的方程(组)须借助优化库 `LuOpt` 中的 `iFind`、`Find` 和 `Opt` 函数求解。例子多来自于互联网,部分例子难度较大,例子 4~7 使用 Matlab、Mathematica、Maple 等软件不易获得正解。

(1) `math::btFindRoot (f,a,b,h,k,eps)`: 对分法求非线性方程的实根。

`f`: 自定义一元函数句柄。函数自变量不能重新赋值。

`a,b`: 求根区间的左端点和右端点 ( $b>a$ )。

`h`: 搜索求根时采用的步长 ( $h>0$ )。

`k`: 可能的实根个数。可缺省,缺省值为 10。

`eps`: 控制精度 ( $eps>0$ )。可缺省,缺省值为  $1e-6$ 。

返回值: 返回一个数组(存放搜索到的实根),或者返回 `nil`。若返回数组的长度为 `k`,则有可能在求根区间 `[a,b]` 内的实根未搜索完。

例子 1: 解方程  $f(x) = x^6 - 5x^5 + 3x^4 + x^3 - 7x^2 + 7x - 20$ ;

Lu 代码:

```
f(x)=x^6-5*x^5+3*x^4+x^3-7*x^2+7*x-20; //函数定义
math::btFindRoot[@f,-2,5,0.2,0]; //函数 o 用于输出一个对象
结果:
```

```
-1.40246 4.33376
```

(2) `math::netn (f,x,eps,t,h,k)`: 求非线性方程组一组实根的拟牛顿法。

`f`: 自定义二元或  $2n$  ( $n>1$ ) 元函数句柄,由用户自编。

如果 `f` 是二元函数,则两个参数为等长的一维数组;

`f(x,y)=` //二元函数,`x[i]`为自变量,`y[i]`为方程右端点函数值

```
{
    y[0]=f1(x[0],x[1],...,x[n-1]),
    y[1]=f2(x[0],x[1],...,x[n-1]),
    ... ..
    y[n-1]=fn(x[0],x[1],...,x[n-1])
};
```

或者:

`f(x1,x2,...,xn,y1,y2,...,yn)=//2n 元函数,xi 为自变量,yi 为方程右端点函数值`

```
{
    y1=f1(x1,x2,...,xn),
    y2=f2(x1,x2,...,xn),
    ... ..
    yn=fn(x1,x2,...,xn)
};
```

`x`: 双精度实型一维数组,长度不小于 `n`,存放一组初值,返回方程组的一组实根。

`eps`: 控制精度 ( $eps>0$ )。可缺省该参数,缺省值  $eps=1e-6$ 。

`t`: 控制 `h` 大小的变量,  $0<t<1$ 。可缺省该参数,缺省值  $h=0.1$ 。

`h`: 增量初值,在本函数中将被破坏。可缺省该参数,缺省值  $h=0.1$ 。

`k`: 允许的最大迭代次数。可缺省该参数,缺省值  $k=100$ 。

返回值: 返回值为实际迭代次数。若返回值为 -1 或 -2 表示求解失败,但这两种情况可放宽精度要求、改变各个初值或改变各个方程顺序再试;若返回值等于 0 说明迭代了 `k` 次仍未满足精度要求,程序工作失败。

例子 2: 解方程组

$$x_1^2 + x_2^2 + x_3^2 - 1.0 = 0$$

$$2.0 \cdot x_1 \cdot x_2 + x_2^2 - 4.0 \cdot x_3 = 0$$

$$3.0 \cdot x_1 \cdot x_2 - 4.0 \cdot x_2 + x_3^2 = 0$$

Lu 代码 1:

```
!!! using["math"]; //使用命名空间 math
```

```
f(x1,x2,x3,y1,y2,y3)=//函数定义
```

```
{
    y1=x1*x1+x2*x2+x3*x3-1.0,
    y2=2.0*x1*x2+x2*x2-4.0*x3,
    y3=3.0*x1*x2-4.0*x2+x3*x3
};
```

```
main(:x,i)={
```

```
    //申请一维数组并赋初值的常规方法。函数 new 表示
    //申请一个对象,real_s 表示数组对象,data 后的数据为初值
    //1.0,1.0,1.0
```

```
    x=new(real_s,data:1.0,1.0,1.0),
```

```
    i=netn[@f,x], //拟牛顿法解方程
```

```
    x.outa(), //用函数 outa 输出一个数组
```

```
    i //返回迭代次数
```

```
};
```

Lu 代码 2:

```
!!! using["math"];
```





## TROUBLESHOOTING OF PROGRAM

```
f(x,y)=
{
    y[0]=x[0]*x[0]+x[1]*x[1]+x[2]*x[2]-1.0,
    y[1]=2.0*x[0]*x[0]+x[1]*x[1]-4.0*x[2],
    y[2]=3.0*x[0]*x[0]-4.0*x[1]+x[2]*x[2]
};
main(:x,i)={
    //申请一维数组并赋初值的简便方法。函数 ra1(realarray1
//的缩写)表示申请一维数组,初值为 1,1.0,1(ra1 自动将整数转
//换为实数)
```

```
i=netn[@f,x=ra1(1,1.0,1)],
x.o(),//用函数 o 输出一个对象
i
};
```

结果:

```
0.785197 0.496611 0.369923
4
```

(3) `luopt::iFind (f, luopt::optmax,m, luopt::optrange,min,max, luopt::opteps,eps, luopt::optpara,x1,x2,..., luopt::optthis,i)`: 求单变量方程的全部解。

f: 自定义 n 元函数句柄, 不可缺省。格式如下:

```
f(x1,x2,...,xn)=
{
    ... ..
};
```

默认求方程  $f(x_1, x_2, \dots, x_n) = 0$  第一个自变量的全部解, 而其他自变量赋值为 0.0。可以用参数 `optthis` 指定求解的自变量, 也可以用参数 `optpara` 给出其他自变量的值。

`luopt::optmax,m`: 区间分割数目 (大于等于 10), 区间分割数目越多精度越高。可以缺省该参数, 缺省值为 200。

`luopt::optrange,min,max`: 指定求解区间。若缺省该参数, 则 `min` 为  $-1e50$ , `max` 为  $1e50$ 。

`luopt::opteps,eps`: 控制精度要求,  $eps > 0$ 。可以缺省该参数, 缺省值为  $1e-6$ 。

`luopt::optpara,x1,x2,...`: 给指定求解自变量之外的其他自变量赋值, 参数 `x1,x2,...` 的个数比全部自变量个数少 1。若缺省该参数, 其他自变量缺省值均为 0.0。

`luopt::optthis,i`: 指定求解的自变量。0 表示第一个自变量; 1 表示第二个自变量, 以此类推。若缺省该参数, `i` 为 0。

返回值: 解的个数。

例子 3: 求方程的全部实数解  $f(x) = x^6 - 5x^5 + 3x^4 + x^3 - 7x^2 + 7x - 20$ ;

Lu 代码:

```
f(x)=x^6-5*x^5+3*x^4+x^3-7*x^2+7*x-20;
luopt::iFind[@f];
```

结果 (解, 误差):

```
-1.402463030422577 3.552713678800501e-015
```

```
4.333755446919994 -2.984279490192421e-013
2
```

例子 4: 解方程  $f(A, B, x) = 8.314 * 298 * [\ln(1-x) + 0.999 * x + A * x^2] - B$ ;

已知 A 和 B 为等长的数组 (原题数组长度为 1000, 本例为了简单, 数组长度取 3), 求 x 的值, 需要得到方程的所有实数解。A 和 B 如下:

```
A=[1.23, 3.5, -7.8]
B=[10, 0.9, -5.6]
```

Lu 代码:

```
!!! using["luopt","math"]; //使用命名空间 luopt 和 math
f(A,B,x)=8.314*298*[ln(1-x)+0.999*x+A*x^2]-B;
main(:A,B,i)=
{
```

```
A=ra1[1.23, 3.5, -7.8],
B=ra1[10, 0.9, -5.6],
i=0, while(i<3,
```

`iFind[@f, optthis:2, optpara:A(i),B(i)]`, //求单变量方程 //的全部解

```
i++
}
```

```
};
```

结果 (A,B,x,误差):

```
1.23 10. -7.255501445630316e-002 3.5527136
78800501e-015
1.23 10. 7.648966409866478e-002 -1.7763568
39400251e-014
3.5 0.9 -1.083215066983416e-002 -8.637535
131583718e-014
3.5 0.9 1.117896003805153e-002 1.221245327
087672e-015
-7.8 -5.6 -1.656801532268951e-002 2.575717
417130363e-014
-7.8 -5.6 1.643659645253869e-002 -2.6645352
59100376e-015
```

(4) `luopt::Find (f, luopt::optmode,mode, luopt::optrange,x1min,x1max,x2min,x2max,...,xnmin,xnmax, luopt::opteps,eps)`: 求方程组的全部解。

f: 自定义 2\*n 元函数句柄, 不可缺省。格式如下:

```
f(x1,x2,...,xn,y1,y2,...,yn)=
{
    y1= ...,
    y2= ...,
    ... ..,
    yn= ...
};
```

`luopt::optmode,mode`: 工作模式, 取 0、1、2、3、...。通常, 工作模式取值越大, 搜到的解越多, 但耗时越长。若缺省该参数, 工作模式取 0。

luopt::optrange,x1min,x1max,x2min,x2max,...,xnmin,xnmax:

指定求解区间。若缺省该参数,则所有变量区间均为  $[-1e50, 1e50]$ 。

luopt::opteps,eps: 控制精度要求,  $\text{eps} > 0$ 。可以缺省该参数,缺省值为  $1e-6$ 。满足  $\text{sqrt} [(y1*y1+y2*y2+...+yn*yn) / n] < \text{eps}$ 。

返回值: 解的个数。

说明: 该函数的解是不稳定的,需要多次运行该函数以获得需要的解。

例子 5: 解含积分方程组

$$q \int_0^{0.01} e^{-\left(\frac{x}{p+q}\right)^2} dx = 0.0019$$

$$q \int_0^{0.015} e^{-\left(\frac{x}{p-q/100}\right)^3} dx = 0.0029$$

Lu 代码:

```
!!! using["luopt","IMSL"]; //使用命名空间 luopt 和 IMSL
s(x::p,q)=exp{-[(x/(p+q))^2]}; //函数定义
t(x::p,q)=exp{-[(x/(p-q/100))^3]}; //函数定义
f(pp,qq,y1,y2::p,q)=
{
    p=pp, q=qq,
    y1=q*QDAGS[@s,0.,0.01]-0.0019, //QDAGS 是
//FcIMSL 库中的积分函数
    y2=q*QDAGS[@t,0.,0.015]-0.0029
};
Find[@f,optmode:5];
结果 (p,q, 误差):
-0.1493489971586127 0.1932862166102974 0.
-3.572749657695181e-002 0.1902652306720847
1.533293416683374e-019
-0.2370257898202487 0.1933213782071417
3.428548309156935e-019
3
```

(5) luopt::Opt (f, optwaysimdeep, optwayconfra, ... ..): 求无约束或约束条件下的  $n$  维极小值,可用于解方程 (组)。

f: 自定义  $n$  元函数句柄,用于计算目标函数  $f(x1,x2,...,xn)$  的值,不可缺省。该函数由用户自编,格式如下:

```
f(x1,x2,...,xn)=
{
    ... //函数必须返回一个值作为目标函数值
};
```

optwaysimdeep, optwayconfra: 当求解难度较大时使用这两个参数。

Opt 函数的其他参数,这里不做介绍。

返回值: 极小值个数。

说明: 该函数使用随机算法搜索全局最小值,故解是不稳

定的,应多次搜索甚至变换参数搜索,以最小者为最优。

有些方程 (组) 用常规方法求解比较困难,须借助于优化函数 Opt 求解。

例子 6: 解方程组

$$x1^2+x2^2-5*x1*x2*x3=85$$

$$x1^3*x2^3*x3^3*x2=60$$

$$x1^3*x3+x3^3*x1-x2=0.55$$

Lu 代码:

```
f (x1,x2,x3) =(x1^2 +x2^2 -5*x1*x2*x3 -85)^2 +
(x1^3*x2^3*x3^3*x2-60)^2+(x1^3*x3+x3^3*x1-x2-0.55)^2;
luopt::Opt[@f];
```

结果 (x1,x2,x3, 误差):

```
19390.88419484747 0.4499999999999999 2.33057529
4810503e-025 8.287969885478255e-029
```

例子 7: 解方程组

$$a*\exp(-b*7.85)+c*\exp(-d*7.85)=28.9$$

$$-a/b*\exp(-b*7.85)-c/d*\exp(-d*7.85)=500$$

$$a/b^2*\exp(-b*7.85)+c/d^2*\exp(-d*7.85)=233$$

$$a*\exp(-b*1.85)+c*\exp(-d*1.85)=20.9$$

Lu 代码:

```
!!! using["luopt"];
f(a,b,c,d:f1,f2,f3,f4)=
f1=a*exp(-b*7.85)+c*exp(-d*7.85)-28.9,
f2=-a/b*exp(-b*7.85)-c/d*exp(-d*7.85)-500,
f3=a/b^2*exp(-b*7.85)+c/d^2*exp(-d*7.85)-233,
f4=a*exp(-b*1.85)+c*exp(-d*1.85)-20.9,
sqrt[(f1*f1+f2*f2+f3*f3+f4*f4)/4]; //目标函数值
Opt[@f, optwaysimdeep, optwayconfra];
```

多次运行,可得以下 2 组解 (a,b,c,d, 误差):

```
19.0817 6873960328 -5.36612020972517e-002 -
0.1719391618522963 -4.244947571876434e-003 1.700948
63414172e-013
-0.1719391618522608 -4.244947571876049e-003
19.08176873960331 -5.366120209725268e-002 7.536443
80168212e-015
```

(作者: 王禄 冯志广 吴阳 潘辉)

## Q 怎样处理 C# 开发中常见问题

A 这几年用 C# 进行应用程序的开发。开发了一些成功的产品。在开发过程中有成功的喜悦,也有因 bug 难以查找和修改。现在把以前的经验和教训总结以下,希望可以帮助朋友有所收益。

(1) 字符和数字类型转换处理。

建议采用 double.TryParse 代替 Convert.ToDouble(),或者 double.Parse()。与此同时,int 和 float 等类型转换可以类似转换。

演示程序如下:





## TROUBLESHOOTING OF PROGRAM

```
double value = 0;
string text = "123";
//不推荐的用法, text 包含非数字下面语句会引发异常
//Value = double.Parse(text);
//推荐用法
bool isDouble = double.TryParse(text, out value);
if (isDouble == true)
{
    //有效情况的处理
}
Else
{
    //异常情况的处理
}
```

(2) 对象的转换处理。

建议采用 as 来进行类型转换, 不要采用强制类型转换。

演示程序如下:

```
//不推荐的用法, 下面语句有可能引发异常:
//ToolStripMenuItem item = (ToolStripMenuItem)sender;
//推荐用法:
ToolStripMenuItem item = sender as ToolStripMenuItem;
//处理 item 为空的情况
```

(3) 对于关键代码段, 需要考虑 lock 锁, 否则有可能出现异常。

写代码的时候, 有时候需要处理一些耗时或者需要保证一致性的代码, 这个时候需要使用 lock。lock 确保当一个线程位于代码的临界区时, 另一个线程不进入临界区。

演示程序如下:

```
lock(this)
{
    //需要处理的代码
}
```

(4) 获取文件路径的处理。

如果在一个小程序中, 获取相对路径可以通过 Application.StartupPath 来处理, 如果在一个大系统或者考虑复用性的情况下, 例如在 dll 中, 则建议通过获取在其中定义指定类的当前加载的程序集的路径方式处理。

演示程序如下:

```
//不推荐的用法, 使用范围窄, 在 dll 中下面的用法会出现异常。
Application.StartupPath
//推荐用法
// classname 为所调用的类名称
String szPath = Assembly.GetAssembly (typeof(classname)).
Location;
FileInfo fileInfo = new FileInfo(szPath);
if (fileInfo.Exists == true)
{
    szPath = fileInfo.DirectoryName;
}
```

(5) 函数错误提示信息的处理。

除了界面层的函数可以考虑通过提示对话框来处理错误信息, 一般建议通过返回错误编码, 来让调用者自己处理。这样调用者可以根据需要进行灵活多变的处理, 而不是被动地用给出不合理的提示。

演示程序如下:

```
//不建议的用法
// 功能: 测试函数错误信息提示
Void testfun(int x, int y)
{
    if (x == 0)
    {
        MessageBox.Show("x 不能为零");
        Return;
    }
}
//建议用法
// 功能: 测试函数错误信息提示
//备注: 返回 0, 是参数有误, x 不能为零, 正常返回 1
int testfun(int x, int y)
{
    if (x == 0)
    {
        Return 0;
    }
    Return 1;
}
```

以上是多年来经验总结和教训。并用上面推荐方法进行开发, 减少了不少 bug 和错误, 提高了软件的质量和开发进度。

(作者: 张伟强)

## 微软下一代操作系统 Windows Blue 今夏上市

据TheVerge报道, 熟悉微软计划的消息人士透露, 微软正在积极筹备下一代 Windows 操作系统, 尽量使这款开发代号为 Windows Blue 的操作系统在今夏与消费者见面。据消息, Windows Blue 已取得里程碑式的进展, 预计不久的将来将发布一个公众预览版。

综合各方面的消息, Windows Blue 将在未来几个月内推出, 目前已经有部分 Windows 8 用户获得了该系统的测试资格, 相关用户正在进行测试并提供反馈, 为即将正式发布的微软 Windows Blue 做准备。另据我们了解, 微软的在 Windows Blue 中但最大的变化之一是改善了系统搜索功能。

Windows Blue 系统中 Bing 将大展手脚。Windows Blue 团队与 Bing 的团队紧密合作, 将使 Windows Blue 的搜索效率不断提高。许多搜索场景都是有针对性的, 包括为用户搜索一部电影和应用程序的内容。Windows Blue 将支持 7-8 英寸设备并内置 Internet Explorer 11。

## 电脑系统维护经验与技巧

## ? 怎样解决西数硬盘的 C1 值偏高问题

! 西数硬盘 C1 值过高, 并且引起硬盘卡顿的现象, 这是西数硬盘的智能电源管理 (APM) 所导致的。简单来讲, 就是系统在不读取硬盘数据的时候, 西数的硬盘磁头会自动收回去, 当要再次读取的时候硬盘磁头又搭接到盘片上, 重新读取数据, 所以就出现了硬盘吱吱响的问题, 以及玩游戏卡顿现象。这里需要注意几点:

(1) 如果只是单纯的 C1 值高, 没有发现其他副作用, 可以不用担心。

(2) 如果担心将来 C1 达到一定值会出现各种问题, 那么可使用软件 quietHDD 或者用 widle3 修改硬盘参数, 这样 C1 值就不会增加很快。

(3) 如果已经出现了些卡顿的现象, 那么就使用 quietHDD 或者用 widle3 修改, 应该中可以得到解决, 解决不了的问题, 那基本上与之无关。

## ? 硬盘逻辑坏道可以修复吗

! 硬盘坏道分为物理坏道和逻辑坏道两种。逻辑坏道对硬盘影响不大, 可以通过用 Windows 自带的磁盘工具进行修复。进入“我的电脑”中, 选择有逻辑坏道的硬盘, 单击鼠标右键, 选择“属性”→“工具”→“开始检查”就弹出“磁盘扫描程序”, 选中“完全”并将“自动修复错误”打上钩, 单击“开始”就开始对该分区进行扫描和修复。此外, 俄罗斯 Maysoft 公司推出的 MHDD 软件功能非常强大, 也很好用。如果是物理坏道, 这就说明是硬盘磁道产生了物理损伤, 建议用户立即备份资料, 联系售后送修。

## ? 如何处理偶尔开机找不到硬盘的问题

! 应该先检查一下是不是接口接触不良引起的, 比如说电源接口氧化、SATA 线接触不良等, 只需把连接线插紧就可以了。如果电源功率不够, 而且负载能力比较差的话, 在冷启动时很多可能会导致硬盘不能正常工作。

## ? 如何处理移动硬盘无法打开的问题

! 首先要排除供电不足的因素, 把连接线上的两个接口都插上, 而且尽量插到机箱背面的 USB 接口上。这个移动硬盘如果是新买的, 很可能没有格式化, 用户只要新建磁盘分区就行了。选择“我的电脑”→“管理”→“磁盘管理”, 查看

这里是否有一个没有盘符的硬盘, 在上面点右键, 选择新建卷就可以了, 再进行格式化, 之后应该就可以了。

## ? 怎样处理硬盘分区表错误造成无法开机的问题

! 如果电脑启动后, 总是出现“Error5: Error finding VFLOPPY.SYS 和 Error8: Fake floppy driver not found”的提示, 需要按任意键才能继续进行, 可能是软驱或硬盘分区表错误造成的。首先进入 BIOS 中, 找到 Floppy 或 Driver A 之类的选项, 并设置“Disabled”, 便可以屏蔽软驱。如果故障依旧的话, 则建议进入 DOS 界面, 用“Fdisk/mbr”命令重建硬盘分区表, 问题应该就可以解决了。

## ? 怎样解决加载 IDE 硬盘后 CPU 温度升高的问题

! 如果采用了体积小巧的迷你机箱和塔式侧吹散热器, 两者对于机箱内部的风道都有着比较严格的要求。而加装 IDE 硬盘, 则采用了较宽的数据线, 所以就影响了机箱内部的空气流通, CPU 温度自然就会升高。实际上, 如果温度升高不多的话, 还是完全不用在意的。如果温度升高了很多的话, 则建议使用改装后的 IDE 数据线, 或者直接去掉加装的 IDE 硬盘即可。

## ? 如何处理移动硬盘发热量极大的问题

! 如果这块硬盘以前发热量就很大的话, 那么就需要改善散热条件。建议用户花二三十元购买一个 USB 小台扇, 而不是直接在硬盘上加装风扇。这样就可以避免风扇的共振问题, 同时散热条件也可以获得改善。而如果这块移动硬盘在以前使用时并没有出现发热量极大的问题, 那么就是主板 USB 供电出现了问题。如果是主板问题, 就只能尽快送修了, 否则长时间使用将会烧毁 USB 移动硬盘。

## ? 如何处理出现“HDD Controller Failure”的提示的问题

! 如果电脑在自检时总是停留很长时间, 然后屏幕上出现“HDD Controller Failure”的提示, 造成这种故障的原因一般是硬盘数据线接口接触不良, 只要将数据线拔下来重新进行连接即可。另外, 数据线受损也可能出现这种故障, 可采用更换数据线的方法来解决。

## ? 怎样处理硬盘噪音增大的问题

! 如果发现使用硬盘噪音比以前更大了, 也有不规则的





## MAIL TO THE DOCTOR

噪音问题。这未必就是坏道问题。很有可能是多任务、高负载的应用环境所造成的，也可能是系统中的垃圾文件造成的。建议用户重新安装系统，并使用 HD Tune 这款软件来检查硬盘的“健康状况”以及进行“错误扫描”。如果一切正常，但噪音仍然很大的话，也可使用硬盘厂商的降噪软件。希捷、西数、日立这3大厂商都提供了相应的软件，用户不妨从官网下载并试用一下。

❓ 怎样处理使用 OCZ 固态硬盘无法安装 Windows 7 的问题

❗ 使用 OCZ 固态硬盘，在安装 Windows 7 系统时却出现了 STOP 错误，而用普通硬盘安装 Windows 7 系统并不会出现这个问题。这是 OCZ 固态硬盘极为罕见的 Bug，在使用某些 P67 芯片组主板并安装 Windows 7 系统时就会出现这样的错误。建议用户更新至刚刚发布的 2.06 版硬盘固件，问题应该就可以解决了。值得说明的是，更新这款固件还可提升常用应用程序的启动速度，并能加强电源管理，控制空闲模式下的电力消耗。因此，对于使用 OCZ 固态硬盘的用户来说，升级固件的好处还是很多的。

❓ 怎样处理使用两块硬盘导致无法进入系统的问题

❗ 如果电脑增加到了两块硬盘，一拖二电源线供电，经常会出现开机进不了系统的现象。有可能在增加硬盘后，两块硬盘用一拖二电源线供电，导致该线路电流过大，电压偏低，引起 PG 电压过低，低于 4.75V。PG 信号消失，就会造成电脑死机，无法进入系统。因此用户可以试着使用两根电源线分别为硬盘供电，故障应该就可以解决了。如果上述方法无法排除故障，也有可能是电源供电不足的问题，建议更换一个额定功率更大的电源。

❓ 硬盘出现“S.M.A.R.T.”故障怎么处理

❗ S.M.A.R.T.是目前硬盘普遍采用的“自我检测分析及报告技术”该技术通过在硬盘固件中植入特定功能，对硬盘电路、电机、磁头、存储介质等进行监测分析，并将相关信息记录在特定的区域，检测出问题时还会向用户发出警报，以减少数据丢失带来的损失。该功能出现问题便会导致“S.M.A.R.T. BAD”故障，可在 BIOS 中关闭 S.M.A.R.T.选项，以便去除开机时的错误提示。不过此硬盘也许已经出现问题了，应该趁着还能读写，尽快转移数据，并将硬盘送修。希捷硬盘如果用 seatools 工具检测 S.M.A.R.T.失败即可保修，建议用户带上硬盘去找配机的商家，或直接联系希捷的服务中心。

## UIPower 助力新奥特完成软件界面的革新项目

新奥特视频技术有限公司是国内领先的数字媒体内容生产及运营的技术、服务提供商。其自主创新的各类产品及解决方案在各领域都有非常广泛的应用。技术产品包括：图文创作系统、非线性编辑系统、网络制播系统、虚拟演播系统等。应用行业遍及广播电视、党政机关、国防军队、交通能源、航空航天、教育培训、影视公司、企业集团等。典型的客户有中央电视台、北京电视台、上海电视台等。

七年前，各个软件企业都在追求软件功能的开发以解决从无到有的问题，大家都在比谁的软件功能更加强大，所以在添加功能时一般很少在意软件的用户体验。然而，最近的这几年软件用户不仅只是关注软件功能的多寡，而且更看重软件的易用性和可观度，因为毕竟再强大的功能都需要通过软件界面来进行呈现的。新奥特与客户接触最多的市场部和产品部纷纷接到客户对软件产品界面不满意的投诉和抱怨。董事长郑福双敏锐地感觉到软件产品的用户体验度的提升刻不容缓，软件界面一方面会直接影响到用户对软件的上手快慢与工作的效率，另一方面会影响到售前营销活动中的用户第一印象。于是成立软件界面革新项目小组，由公司技术总经理和市场总经理联合牵头，专门负责全公司上下的所有软件产品的界面革新工作。万事开头难，那么多系统，那么多代码，每次组织技术人员讨论界面改造方案时，老总们收到

的都是些反对的意见。大家都担心代码量太大而存在很大的风险。然而，市场的需求摆在眼前，如何快速而顺利地完成本次重大的革新则显得至关重要。郑福双当机立断下了决策：借助外脑。

经过一段时间的多家界面供应商的选择，最终选定了上海勇进软件（UIPower.com）来承担此次所有生产线软件产品的界面革新工作。选择的理由是 UIPower 提供从 UI 的交互设计、视觉设计、界面开发到界面集成的整套流程的整体界面解决方案。目前国内做 UI 设计的公司居多，但他们都无法将 UI 设计真正做到软件程序中去，只能交给软件企业自行完成。而软件界面开发是一项专业要求高，工作量巨大的工作，要彻底做好界面必须借助界面工具，而国内能将工具和服务融为一体的，也只有 UIPower 了。

UIPower 经过三个月的努力，终于将新奥特的一款最核心最庞大的系统—喜马拉雅非线性编辑软件的界面改造完成了。

新奥特自发布带有新界面的喜马拉雅系统以来，获得了来自用户的一致好评，也获得了市场人员的在工作一线的客户良好的反馈。

UIPower 的 DirectUI 产品以界面与业务逻辑的彻底分离技术在各种需求的客户中真正实现了“让天下没有难做的界面”的企业使命。







书名: HTML & CSS 设计与构建网站  
ISBN: 978-7-302-31103-4  
定价: 59.80 元  
作者: (美) 达科特 (Duckett, J.) 著

本书一直稳居 Amazon Computers & Technology 领先地位, 不管您设计和建立新网站, 还是想更好地控制现有网站, 都可以在本书的指导下创建出用户友好、令用户赏心悦目的 Web 内容。我们知道, 编码是一项令人望而生畏的工作, 而本书却采用有别于许多传统编程书籍的新颖编排方式, 将使您收到事半功倍的学习效果。

每一页都在短小精悍的示例代码的引导下, 简明直观、直截了当地阐述一个新主题。本书还提供关于如何组织和设计网页的实用信息, 以便帮助您创建充满魅力、易于使用的网站。



书名: PhoneGap 入门经典  
ISBN: 978-7-302-31106-5  
定价: 49 元  
作者: (美) 迈尔 (Myer, T.) 著

本书是一本 PhoneGap 指南。详细介绍 PhoneGap 应用程序的整个创建过程, 包括通讯录、摄像头、媒体文件、存储选项以及更多内容; 展示了为 iOS、Android、BlackBerry、webOS 以及 Symbian 系统安装和配置 PhoneGap 的方法; 温习了事件对象和事件类型; 分享了使用网络、设备以及通知功能进行工作的技术; 为掌握文件系统、Web 数据库以及存储提供重要的指导; 鼓励你充分利用地理定位、罗盘以及加速计的功能; 可以通过贯穿本书的一组练习来实践所学的知识。



书名: HTML5 多媒体开发指南  
ISBN: 978-7-302-31104-1  
定价: 58.00 元  
作者: (美) 布拉特曼 (Bluttman, K.), (美) 科特雷尔 (Cottrell, L.) 著

现在开发人员可以在浏览器中创建出具有丰富交互功能的内容, 并实现跨平台部署。本书对此进行了详细介绍。本书还介绍了如何使用 HTML5 革命性的新特性, 包括原生的<audio>、<video>和<canvas>元素, 使用这些新元素可以取代第三方插件。以及 HTML5 新标记, 比如简化的 doctype 标记, 另外, 还介绍了本地存储、拖放和 geolocation 功能。本书通过详细的代码示例举例说明了所讨论的技术。通过本书的学习, 读者可以使用 HTML5 开发出炫目的多媒体应用和网站。



书名: 数据挖掘: 概念、模型、方法和算法 (第2版)  
ISBN: 978-7-302-30714-3  
定价: 59 元  
作者: (美) 坎塔尔季奇 (Kantardzic, M.) 著

本书介绍了通过分析高维数据空间中的海量原始数据来提取用于决策的新信息的尖端技术和方法。

本书开篇阐述数据挖掘原理, 此后在示例的引导下详细讲解起源于统计学、机器学习、神经网络、模糊逻辑和演化计算等学科的具有代表性的、极前沿的挖掘方法和算法。本书还着重描述如何恰当地选择方法和数据分析软件并合理地调整参数。每章末尾附有复习题。





## 全新上市 汇集世界权威安全经验



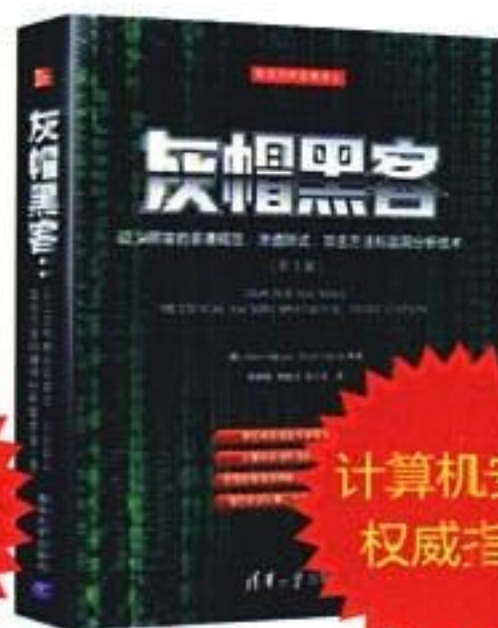
国际密码学会  
会长推荐

书号: 9787302296096  
定价: 59.00元



ebay总裁  
作序推荐

书号: 9787302293781  
定价: 59.00元



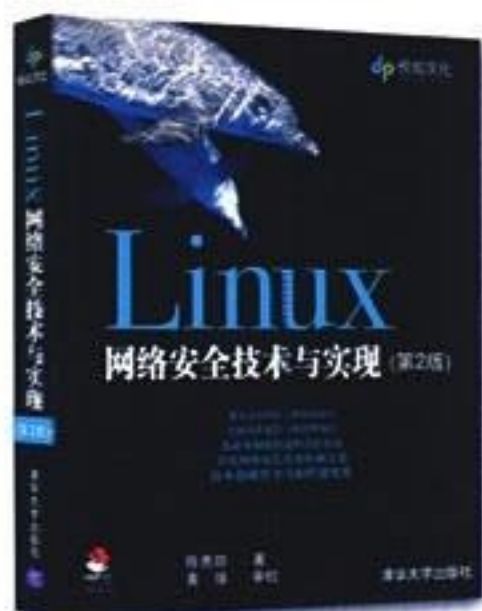
计算机安全  
权威指南

书号: 9787302301509  
定价: 73.80元



"上帝之锤"  
创始人  
全新力作

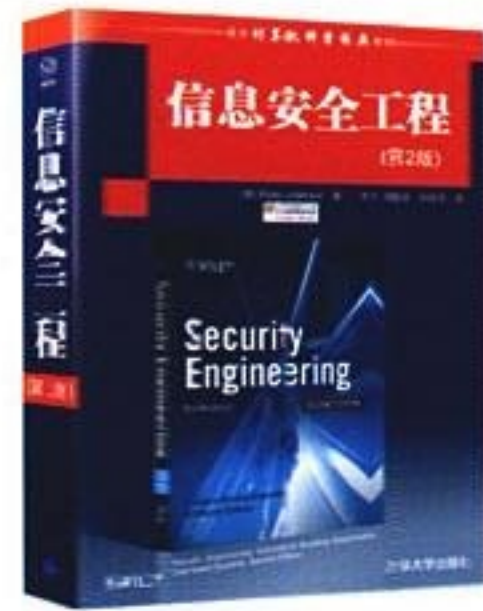
书号: 9787302307433  
定价: 58.00元



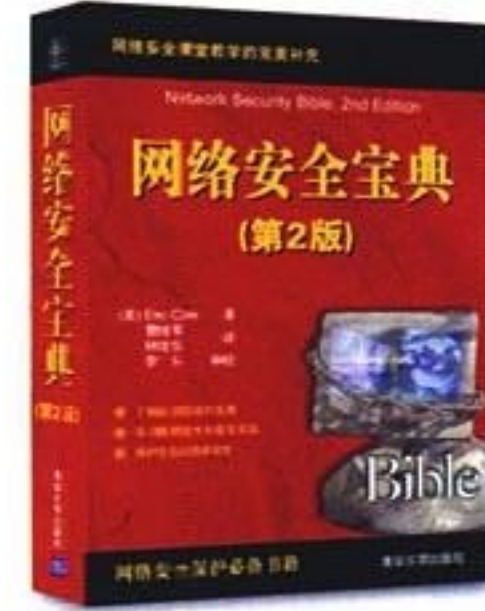
书号: 9787302278863  
定价: 68.00元



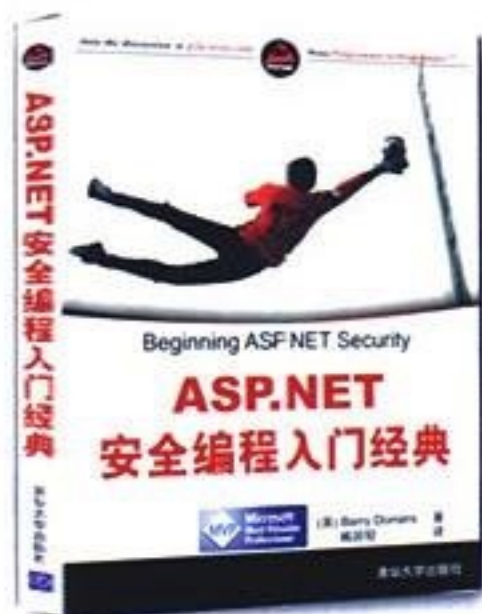
书号: 9787302274407  
定价: 69.80元



书号: 9787302271154  
定价: 80.00元



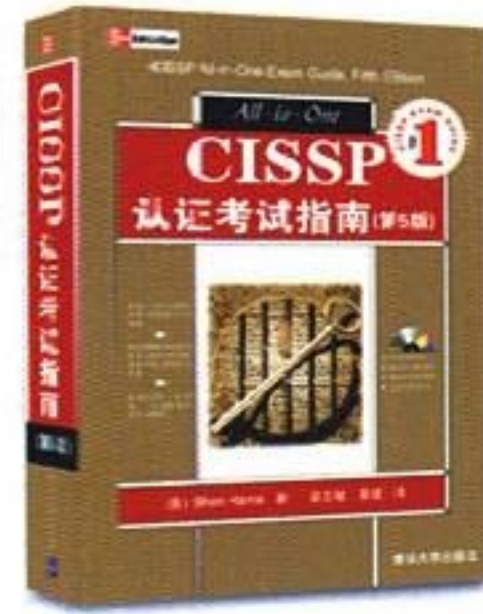
书号: 9787302239390  
定价: 98.00元



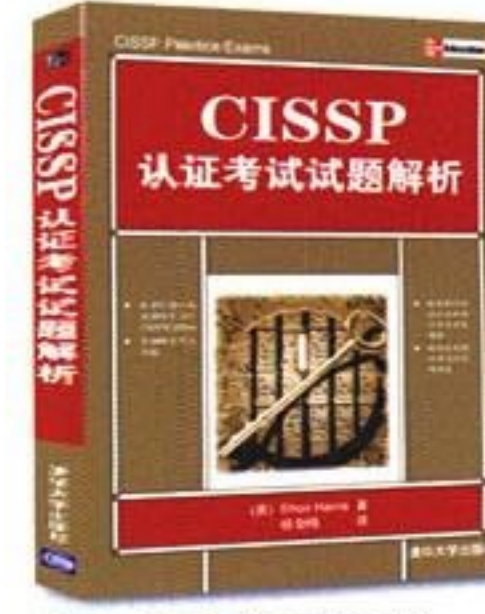
书号: 9787302263746  
定价: 59.80元



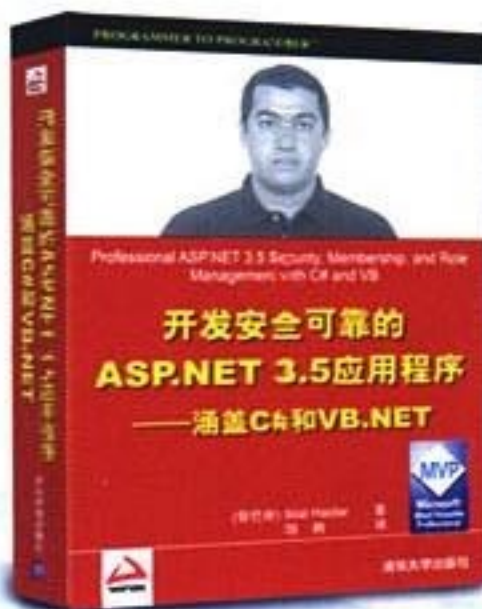
书号: 9787302256328  
定价: 69.00元



书号: 9787302269809  
定价: 133.00元



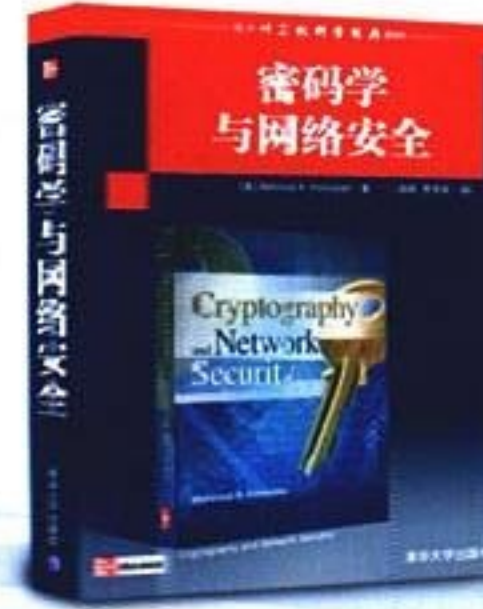
书号: 9787302258018  
定价: 43.00元



书号: 9787302228929  
定价: 118.00元



书号: 9787302226345  
定价: 48.00元



书号: 9787302197270  
定价: 59.90元



书号: 9787302188704  
定价: 86.00元



汇主流编程语言  
聚宝贵编程经验

精选典型编程案例  
揭示编程技术诀窍



精华版



《电脑编程技巧与维护》杂志社 编著  
定价: 79元

订阅方式:

汇款地址: 北京市海淀区长春桥路5号6号楼1209室

杂志社官方淘宝店: <http://comprg.taobao.com>

E-mail: [zzsfx@vip.sina.com](mailto:zzsfx@vip.sina.com) QQ: 565693495

汇款如未注明所购买数量和邮寄地址, 请与杂志社联系。

收款人: 电脑编程技巧与维护杂志社 邮编: 100039

电话/传真: 82561614



来卡网出品  
LAICAR.COM  
[shop35833438.taobao.com](http://shop35833438.taobao.com)